m2900

BIPOLAR (TTL) PROCESSOR FAMILY

MOTOROLA Semiconductor Products Inc.

M2900
BIPOLAR (TTL)
PROCESSOR FAMILY

First Edition

Printed in Switzerland

TABLE OF CONTENTS

MC2901

FOUR-BIT BIPOLAR MICROPROCESSOR SLICE

DISTINCTIVE CHARACTERISTICS

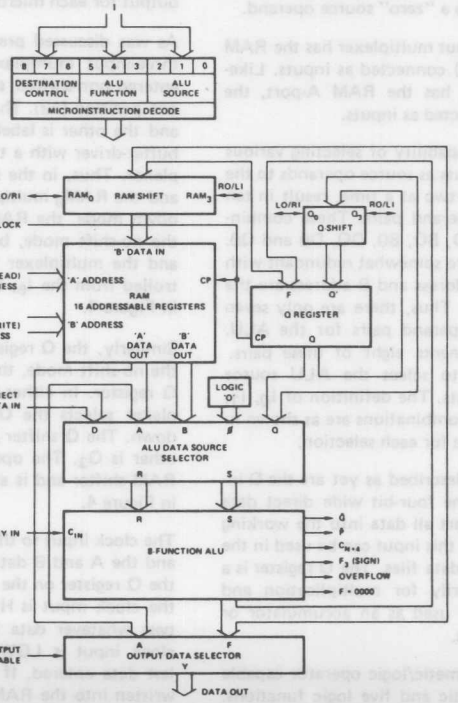
- Two-address architecture –
  Independent simultaneous access to two working registers saves machine cycles.
- Eight-function ALU –
  Performs addition, two subtraction operations, and five logic functions on two source operands.
- Flexible data source selection –
  ALU data is selected from five source ports for a total of 203 source operand pairs for every ALU function.
- Left/right shift independent of ALU –
  Add and shift operations take only one cycle.
- Four status flags –
  Carry, overflow, zero, and negative.
- Expandable –
  Connect any number of MC2901's together for longer word lengths.
- Microprogrammable –
  Three groups of three bits each for source operand, ALU function, and destination control.

GENERAL DESCRIPTION

The four-bit bipolar microprocessor slice is designed as a high-speed cascadable element intended for use in CPU's, peripheral controllers, programmable microprocessors and numerous other applications. The microinstruction flexibility of the MC2901 will allow efficient emulation of almost any digital computing machine.

The device, as shown in the block diagram below, consists of a 16-word by 4-bit two-port RAM, a high-speed ALU, and the associated shifting, decoding and multiplexing circuitry. The nine-bit microinstruction word is organized into three groups of three bits each and selects the ALU source operands, the ALU function, and the ALU destination register. The microprocessor is cascadable with full look-ahead or with ripple carry, has three-state outputs, and provides various status flag outputs from the ALU. Advanced low-power Schottky processing is used to fabricate this 40-lead LSI chip.

MICROPROCESSOR SLICE BLOCK DIAGRAM



ARCHITECTURE

A detailed block diagram of the bipolar microprogrammable microprocessor structure is shown in Figure 1. The circuit is a four-bit slice cascadable to any number of bits. Therefore, all data paths within the circuit are four bits wide. The two key elements in the Figure 1 block diagram are the 16-word by 4-bit 2-port RAM and the high-speed ALU.

Data in any of the 16 words of the Random Access Memory (RAM) can be read from the A-port of the RAM as controlled by the 4-bit A address field input. Likewise, data in any of the 16 words of the RAM as defined by the B address field input can be simultaneously read from the B-port of the RAM. The same code can be applied to the A select field and B select field in which case the identical file data will appear at both the RAM A-port and B-port outputs simultaneously.

When enabled by the RAM write enable (RAM EN), new data is always written into the file (word) defined by the B address field of the RAM. The RAM data input field is driven by a 3-input multiplexer. This configuration is used to shift the ALU output data (F) if desired. This three-input multiplexer scheme allows the data to be shifted up one bit position, shifted down one bit position, or not shifted in either direction.

The RAM A-port data outputs and RAM B-port data outputs drive separate 4-bit latches. These latches hold the RAM data while the clock input is LOW. This eliminates any possible race conditions that could occur while new data is being written into the RAM.

The high-speed Arithmetic Logic Unit (ALU) can perform three binary arithmetic and five logic operations on the two 4-bit input words R and S. The R input field is driven from a 2-input multiplexer, while the S input field is driven from a 3-input multiplexer. Both multiplexers also have an inhibit capability; that is, no data is passed. This is equivalent to a "zero" source operand.

Referring to Figure 1, the ALU R-input multiplexer has the RAM A-port and the direct data inputs (D) connected as inputs. Likewise, the ALU S-input multiplexer has the RAM A-port, the RAM B-port and the Q register connected as inputs.

This multiplexer scheme gives the capability of selecting various pairs of the A, B, D, Q and "0" inputs as source operands to the ALU. These five inputs, when taken two at a time, result in ten possible combinations of source operand pairs. These combinations include AB, AD, AQ, A0, BD, BQ, B0, DQ, D0 and Q0. It is apparent that AD, AQ and A0 are somewhat redundant with BD, BQ and B0 in that if the A address and B address are the same, the identical function results. Thus, there are only seven completely non-redundant source operand pairs for the ALU. The MC2901 microprocessor implements eight of these pairs. The microinstruction inputs used to select the ALU source operands are the I_0, I_1, and I_2 inputs. The definition of I_0, I_1, and I_2 for the eight source operand combinations are as shown in Figure 2. Also shown is the octal code for each selection.

The two source operands not fully described as yet are the D input and Q input. The D input is the four-bit wide direct data field input. This port is used to insert all data into the working registers inside the device. Likewise, this input can be used in the ALU to modify any of the internal data files. The Q register is a separate 4-bit file intended primarily for multiplication and division routines but it can also be used as an accumulator or holding register for some applications.

The ALU itself is a high-speed arithmetic/logic operator capable of performing three binary arithmetic and five logic functions. The I_3, I_4, and I_5 microinstruction inputs are used to select the ALU function. The definition of these inputs is shown in Figure 3. The octal code is also shown for reference. The normal technique for cascading the ALU of several devices is in a look-ahead carry mode. Carry generate, $\overline{G}$, and carry propagate, $\overline{P}$, are outputs of the device for use with a carry-look-ahead-generator such as the MC2902 ('182). A carry-out, C_{n+4}, is also generated and is available as an output for use as the carry flag in a status register. Both carry-in (C_n) and carry-out (C_{n+4}) are active HIGH.

The ALU has three other status-oriented outputs. These are F_3, F = 0, and overflow (OVR). The F_3 output is the most significant (sign) bit of the ALU and can be used to determine positive or negative results without enabling the three-state data outputs. F_3 is non-inverted with respect to the sign bit output Y_3. The F = 0 output is used for zero detect. It is an open-collector output and can be wire OR'ed between microprocessor slices. F = 0 is HIGH when all F outputs are LOW. The overflow output (OVR) is used to flag arithmetic operations that exceed the available two's complement number range. The overflow output (OVR) is HIGH when overflow exists. That is, when C_{n+3} and C_{n+4} are not the same polarity.

The ALU data output is routed to several destinations. It can be a data output of the device and it can also be stored in the RAM or the Q register. Eight possible combinations of ALU destination functions are available as defined by the I_6, I_7, and I_8 microinstruction inputs. These combinations are shown in Figure 4.

The four-bit data output field (Y) features three-state outputs and can be directly bus organized. An output control ($\overline{OE}$) is used to enable the three-state outputs. When $\overline{OE}$ is HIGH, the Y outputs are in the high-impedance state.

A two-input multiplexer is also used at the data output such that either the A-port of the RAM or the ALU outputs (F) are selected at the device Y outputs. This selection is controlled by the I_6, I_7, and I_8 microinstruction inputs. Refer to Figure 4 for the selected output for each microinstruction code combination.

As was discussed previously, the RAM inputs are driven from a three-input multiplexer. This allows the ALU outputs to be entered non-shifted, shifted up one position (X2) or shifted down one position (÷2). The shifter has two ports; one is labeled RAM_0 and the other is labeled RAM_3. Both of these ports consist of a buffer-driver with a three-state output and an input to the multiplexer. Thus, in the shift up mode, the RAM_3 buffer is enabled and the RAM_0 multiplexer input is enabled. Likewise, in the shift down mode, the RAM_0 buffer and RAM_3 input are enabled. In the no-shift mode, both buffers are in the high-impedance state and the multiplexer inputs are not selected. This shifter is controlled from the I_6, I_7 and I_8 microinstruction inputs as defined in Figure 4.

Similarly, the Q register is driven from a 3-input multiplexer. In the no-shift mode, the multiplexer enters the ALU data into the Q register. In either the shift-up or shift-down mode, the multiplexer selects the Q register data appropriately shifted up or down. The Q shifter also has two ports; one is labeled Q_0 and the other is Q_3. The operation of these two ports is similar to the RAM shifter and is also controlled from I_6, I_7, and I_8 as shown in Figure 4.

The clock input to the MC2901 controls the RAM, the Q register, and the A and B data latches. When enabled, data is clocked into the Q register on the LOW-to-HIGH transition of the clock. When the clock input is HIGH, the A and B latches are open and will pass whatever data is present at the RAM outputs. When the clock input is LOW, the latches are closed and will retain the last data entered. If the RAM-EN is enabled, new data will be written into the RAM file (word) defined by the B address field when the clock input is LOW.

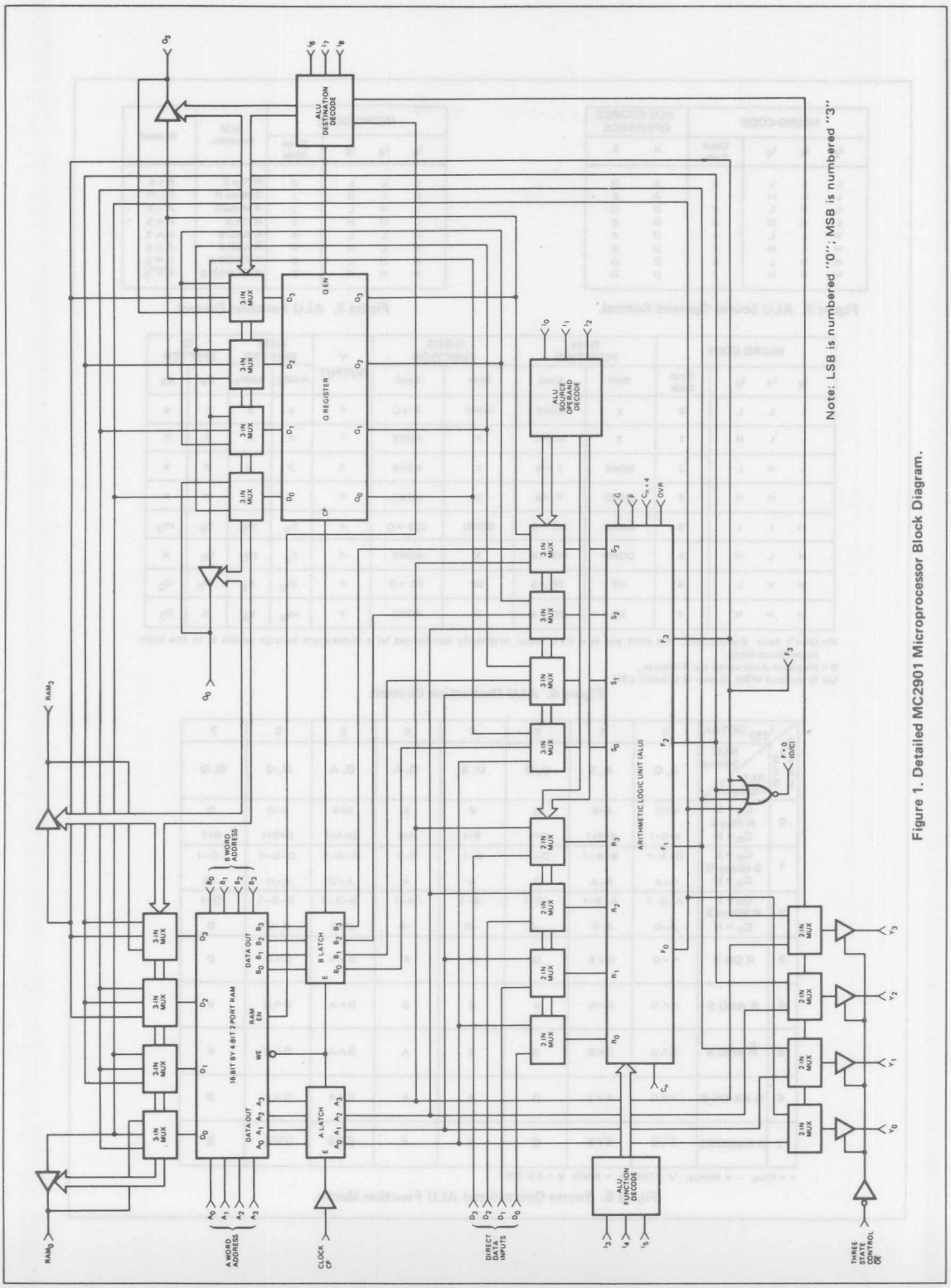


Figure 1. Detailed MC2901 Microprocessor Block Diagram.

MICRO CODE				ALU SOURCE OPERANDS	
I_2	I_1	I_0	Octal Code	R	S
L	L	L	0	A	Q
L	L	H	1	A	B
L	H	L	2	O	Q
L	H	H	3	O	B
H	L	L	4	O	A
H	L	H	5	D	A
H	H	L	6	D	Q
H	H	H	7	D	O

Figure 2. ALU Source Operand Control.

MICRO CODE				ALU Function	Symbol
I_5	I_4	I_3	Octal Code		
L	L	L	0	R Plus S	R + S
L	L	H	1	S Minus R	S − R
L	H	L	2	R Minus S	R − S
L	H	H	3	R OR S	R ∨ S
H	L	L	4	R AND S	R ∧ S
H	L	H	5	$\overline{R}$ AND S	$\overline{R} \wedge S$
H	H	L	6	R EX-OR S	R ∀ S
H	H	H	7	R EX-NOR S	$\overline{R \forall S}$

Figure 3. ALU Function Control.

MICRO CODE				RAM FUNCTION		Q-REG. FUNCTION		Y OUTPUT	RAM SHIFTER		Q SHIFTER	
I_8	I_7	I_6	Octal Code	Shift	Load	Shift	Load		RAM_0	RAM_3	Q_0	Q_3
L	L	L	0	X	NONE	NONE	F→Q	F	X	X	X	X
L	L	H	1	X	NONE	X	NONE	F	X	X	X	X
L	H	L	2	NONE	F→B	X	NONE	A	X	X	X	X
L	H	H	3	NONE	F→B	X	NONE	F	X	X	X	X
H	L	L	4	DOWN	F/2→B	DOWN	Q/2→Q	F	F_0	IN_3	Q_0	IN_3
H	L	H	5	DOWN	F/2→B	X	NONE	F	F_0	IN_3	Q_0	X
H	H	L	6	UP	2F→B	UP	2Q→Q	F	IN_0	F_3	IN_0	Q_3
H	H	H	7	UP	2F→B	X	NONE	F	IN_0	F_3	X	Q_3

X= Don't care. Electrically, the shift pin is a TTL input internally connected to a three-state output which is in the high-impedance state.
B = Register Addressed by B inputs.
Up is toward MSB, Down is toward LSB.

Figure 4. ALU Destination Control.

OCTAL I_{210} / OCTAL I_{543} / ALU Source / ALU Function		0	1	2	3	4	5	6	7
		A, Q	A, B	O, Q	O, B	O, A	D, A	D, Q	D, O
0	C_n = L R Plus S C_n = H	A+Q A+Q+1	A+B A+B+1	Q Q+1	B B+1	A A+1	D+A D+A+1	D+Q D+Q+1	D D+1
1	C_n = L S Minus R C_n = H	Q−A−1 Q−A	B−A−1 B−A	Q−1 Q	B−1 B	A−1 A	A−D−1 A−D	Q−D−1 Q−D	−D−1 −D
2	C_n = L R Minus S C_n = H	A−Q−1 A−Q	A−B−1 A−B	−Q−1 −Q	−B−1 −B	−A−1 −A	D−A−1 D−A	D−Q−1 D−Q	D−1 D
3	R OR S	A∨Q	A∨B	Q	B	A	D∨A	D∨Q	D
4	R AND S	A∧Q	A∧B	0	0	0	D∧A	D∧Q	0
5	$\overline{R}$ AND S	$\overline{A}$∧Q	$\overline{A}$∧B	Q	B	A	$\overline{D}$∧A	$\overline{D}$∧Q	0
6	R EX-OR S	A∀Q	A∀B	Q	B	A	D∀A	D∀Q	D
7	R EX-NOR S	$\overline{A \forall Q}$	$\overline{A \forall B}$	$\overline{Q}$	$\overline{B}$	$\overline{A}$	$\overline{D \forall A}$	$\overline{D \forall Q}$	$\overline{D}$

+ = Plus; − = Minus; ∨ = OR; ∧ = AND; ∀ = EX-OR

Figure 5. Source Operand and ALU Function Matrix.

SOURCE OPERANDS AND ALU FUNCTIONS

There are eight source operand pairs available to the ALU as selected by the I_0, I_1, and I_2 instruction inputs. The ALU can perform eight functions; five logic and three arithmetic. The I_3, I_4, and I_5 instruction inputs control this function selection. The carry input, C_n, also affects the ALU results when in the arithmetic mode. The C_n input has no effect in the logic mode. When I_0 through I_5 and C_n are viewed together, the matrix of Figure 5 results. This matrix fully defines the ALU/source operand function for each state.

The ALU functions can also be examined on a "task" basis, i.e., add, subtract, AND, OR, etc. In the arithmetic mode, the carry will affect the function performed while in the logic mode, the carry will have no bearing on the ALU output. Figure 6 defines the various logic operations that the MC2901 can perform and Figure 7 shows the arithmetic functions of the device. Both carry-in LOW (C_n = 0) and carry-in HIGH (C_n = 1) are defined in these operations.

Octal I_{543}, I_{210}	Group	Function
4 0	AND	$A \wedge Q$
4 1		$A \wedge B$
4 5		$D \wedge A$
4 6		$D \wedge Q$
3 0	OR	$A \vee Q$
3 1		$A \vee B$
3 5		$D \vee A$
3 6		$D \vee Q$
6 0	EX-OR	$A \forall Q$
6 1		$A \forall B$
6 5		$D \forall A$
6 6		$D \forall Q$
7 0	EX-NOR	$\overline{A \forall Q}$
7 1		$\overline{A \forall B}$
7 5		$\overline{D \forall A}$
7 6		$\overline{D \forall Q}$
7 2	INVERT	$\overline{Q}$
7 3		$\overline{B}$
7 4		$\overline{A}$
7 7		$\overline{D}$
6 2	PASS	Q
6 3		B
6 4		A
6 7		D
3 2	PASS	Q
3 3		B
3 4		A
3 7		D
4 2	"ZERO"	0
4 3		0
4 4		0
4 7		0
5 0	MASK	$\overline{A} \wedge Q$
5 1		$\overline{A} \wedge B$
5 5		$\overline{D} \wedge A$
5 6		$\overline{D} \wedge Q$

Figure 6. ALU Logic Mode Functions. (C_n Irrelevant)

Octal	C_n = 0 (Low)		C_n = 1 (High)	
I_{543}, I_{210}	Group	Function	Group	Function
0 0	ADD	A+Q	ADD plus one	A+Q+1
0 1		A+B		A+B+1
0 5		D+A		D+A+1
0 6		D+Q		D+Q+1
0 2	PASS	Q	Increment	Q+1
0 3		B		B+1
0 4		A		A+1
0 7		D		D+1
1 2	Decrement	Q–1	PASS	Q
1 3		B–1		B
1 4		A–1		A
2 7		D–1		D
2 2	1's Comp.	–Q–1	2's Comp. (Negate)	–Q
2 3		–B–1		–B
2 4		–A–1		–A
1 7		–D–1		–D
1 0	Subtract (1's Comp)	Q–A–1	Subtract (2's Comp)	Q–A
1 1		B–A–1		B–A
1 5		A–D–1		A–D
1 6		Q–D–1		Q–D
2 0		A–Q–1		A–Q
2 1		A–B–1		A–B
2 5		D–A–1		D–A
2 6		D–Q–1		D–Q

Figure 7. ALU Arithmetic Mode Functions.

LOGIC FUNCTIONS FOR G, P, C_{n+4}, AND OVR

The four signals G, P, C_{n+4}, and OVR are designed to indicate carry and overflow conditions when the MC2901 is in the add or subtract mode. The table below indicates the logic equations for these four signals for each of the eight ALU functions. The R and S inputs are the two inputs selected according to Figure 2.

Definitions (+ = OR)

$P_0 = R_0 + S_0$ $\quad G_0 = R_0S_0$
$P_1 = R_1 + S_1$ $\quad G_1 = R_1S_1$
$P_2 = R_2 + S_2$ $\quad G_2 = R_2S_2$
$P_3 = R_3 + S_3$ $\quad G_3 = R_3S_3$

$C_4 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_n$

$C_3 = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_n$

I_{543}	Function	$\overline{P}$	$\overline{G}$	C_{n+4}	OVR
0	R + S	$\overline{P_3P_2P_1P_0}$	$\overline{G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0}$	C_4	$C_3 \forall C_4$
1	S − R	← Same as R + S equations, but substitute $\overline{R_i}$ for R_i in definitions →			
2	R − S	← Same as R + S equations, but substitute $\overline{S_i}$ for S_i in definitions →			
3	R ∨ S	LOW	$P_3P_2P_1P_0$	$\overline{P_3P_2P_1P_0} + C_n$	$\overline{P_3P_2P_1P_0} + C_n$
4	R ∧ S	LOW	$\overline{G_3 + G_2 + G_1 + G_0}$	$G_3 + G_2 + G_1 + G_0 + C_n$	$G_3 + G_2 + G_1 + G_0 + C_n$
5	$\overline{R}$ ∧ S	LOW	← Same as R ∧ S equations, but substitute $\overline{R_i}$ for R_i in definitions →		
6	R ∀ S	← Same as $\overline{R \forall S}$, but substitute $\overline{R_i}$ for R_i in definitions →			
7	$\overline{R \forall S}$	$G_3 + G_2 + G_1 + G_0$	$G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1P_0$	$\overline{G_3 + P_3G_2 + P_3P_2G_1} + P_3P_2P_1P_0 (G_0 + \overline{C_n})$	See Note

Note: $[\overline{P}_2+\overline{G}_2\overline{P}_1+\overline{G}_2\overline{G}_1\overline{P}_0+\overline{G}_2\overline{G}_1\overline{G}_0C_n] \forall [\overline{P}_3+\overline{G}_3\overline{P}_2+\overline{G}_3\overline{G}_2\overline{P}_1+\overline{G}_3\overline{G}_2\overline{G}_1\overline{P}_0+\overline{G}_3\overline{G}_2\overline{G}_1\overline{G}_0C_n]$ + = OR

Figure 8.

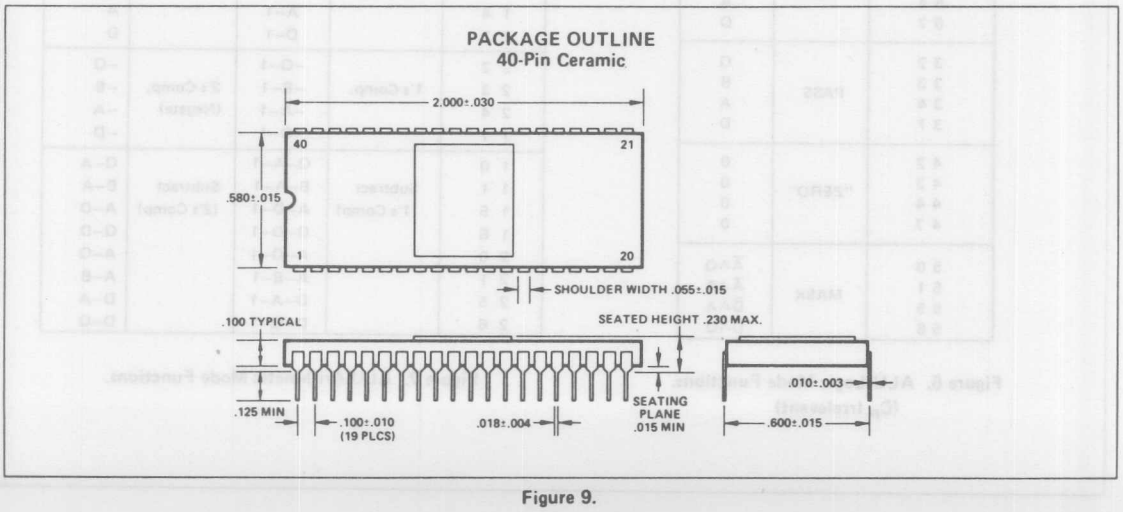


Figure 9.

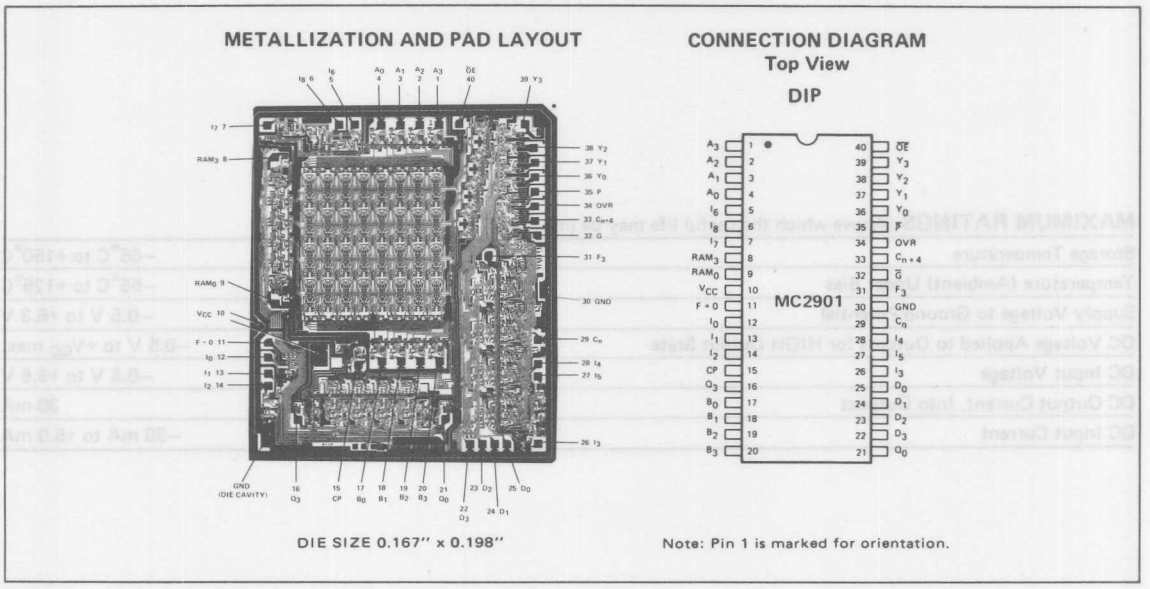


Figure 10.

PIN DEFINITIONS

A_{0-3} The four address inputs to the register stack used to select one register whose contents are displayed through the A-port.

B_{0-3} The four address inputs to the register stack used to select one register whose contents are displayed through the B-port and into which new data can be written when the clock goes LOW.

I_{0-8} The nine instruction control lines to the MC2901, used to determine what data sources will be applied to the ALU (I_{012}), what function the ALU will perform (I_{345}), and what data is to be deposited in the Q-register or the register stack (I_{678}).

Q_3 RAM_3 A shift line at the MSB of the Q register (Q_3) and the register stack (RAM_3). Electrically these lines are three-state outputs connected to TTL inputs internal to the MC2901. When the destination code on I_{678} indicates an up shift (octal 6 or 7) the three-state outputs are enabled and the MSB of the Q register is available on the Q_3 pin and the MSB of the ALU output is available on the RAM_3 pin. Otherwise, the three-state outputs are OFF (high-impedance) and the pins are electrically LS-TTL inputs. When the destination code calls for a down shift, the pins are used as the data inputs to the MSB of the Q register (octal 4) and RAM (octal 4 or 5).

Q_0 RAM_0 Shift lines like Q_3 and RAM_3, but at the LSB of the Q-register and RAM. These pins are tied to the Q_3 and RAM_3 pins of the adjacent device to transfer data between devices for up and down shifts of the Q register and ALU data.

D_{0-3} Direct data inputs. A four-bit data field which may be selected as one of the ALU data sources for entering data into the MC2901. D_0 is the LSB.

Y_{0-3} The four data outputs of the MC2901. These are three-state output lines. When enabled, they display either the four outputs of the ALU or the data on the A-port of the register stack, as determined by the destination code I_{678}.

$\overline{OE}$ Output Enable. When $\overline{OE}$ is HIGH, the Y outputs are OFF; when $\overline{OE}$ is LOW, the Y outputs are active (HIGH or LOW).

$\overline{P}$, $\overline{G}$ The carry generate and propagate outputs of the MC2901's ALU. These signals are used with the MC2902 for carry-lookahead. See Figure 8 for the logic equations.

OVR Overflow. This pin is logically the Exclusive-OR of the carry-in and carry-out of the MSB of the ALU. At the most significant end of the word, this pin indicates that the result of an arithmetic two's complement operation has overflowed into the sign-bit. See Figure 8 for logic equation.

F = 0 This is an open collector output which goes HIGH (OFF) if the data on the four ALU outputs F_{0-3} are all LOW. In positive logic, it indicates the result of an ALU operation is zero.

C_n The carry-in to the MC2901's ALU.

C_{n+4} The carry-out of the MC2901's ALU. See Figure 8 for equations.

CP The clock to the MC2901. The Q register and register stack outputs change on the clock LOW-to-HIGH transition. The clock LOW time is internally the write enable to the 16 x 4 RAM which comprises the "master" latches of the register stack. While the clock is LOW, the "slave" latches on the RAM outputs are closed, storing the data previously on the RAM outputs. This allows synchronous master-slave operation of the register stack.

MAXIMUM RATINGS (Above which the useful life may be impaired)

Storage Temperature	−65°C to +150°C
Temperature (Ambient) Under Bias	−55°C to +125°C
Supply Voltage to Ground Potential	−0.5 V to +6.3 V
DC Voltage Applied to Outputs for HIGH Output State	−0.5 V to +V_{CC} max.
DC Input Voltage	−0.5 V to +5.5 V
DC Output Current, Into Outputs	30 mA
DC Input Current	−30 mA to +5.0 mA

OPERATING RANGE

P/N	Ambient Temperature	V_{CC}
MC2901 LC	0°C to +70°C	4.75 V to 5.25 V
MC2901 LM	−55°C to +125°C	4.50 V to 5.50 V

ORDERING INFORMATION

Package Type	Temperature Range	Order Number
Hermetic DIP	0°C to +70°C	MC2901 LC
Hermetic DIP	−55°C to +125°C	MC2901 LM
Dice	0°C to +70°C	MCC2901 C

ELECTRICAL CHARACTERISTICS OVER OPERATING RANGE (Unless Otherwise Noted)

(Group A, Subgroups 1, 2 and 3)

Parameters	Description	Test Conditions (Note 1)			Min.	Typ. (Note 2)	Max.	Units
V_{OH}	Output HIGH Voltage	V_{CC} = MIN., V_{IN} = V_{IH} or V_{IL}	I_{OH} = −1.6mA Y_0, Y_1, Y_2, Y_3		2.4			Volts
			I_{OH} = −1.0mA, C_{n+4}		2.4			
			I_{OH} = −800μA, OVR, $\overline{P}$		2.4			
			I_{OH} = −600μA, F_3		2.4			
			I_{OH} = −600μA $RAM_{0,3}, Q_{0,3}$		2.4			
			I_{OH} = −1.6mA, $\overline{G}$		2.4			
I_{CEX}	Output Leakage Current for F = 0 Output	V_{CC} = MIN., V_{OH} = 5.5V V_{IN} = V_{IH} or V_{IL}					250	μA
V_{OL}	Output LOW Voltage	V_{CC} = MIN., V_{IN} = V_{IH} or V_{IL}	I_{OL} = 16mA $Y_0, Y_1, Y_2, Y_3, \overline{G}$				0.5	Volts
			I_{OL} = 10mA, C_{n+4}, F=0				0.5	
			I_{OL} = 8.0mA, OVR, $\overline{P}$				0.5	
			I_{OL} = 6.0mA, F_3 $RAM_{0,3}, Q_{0,3}$				0.5	
V_{IH}	Input HIGH Level	Guaranteed input logical HIGH voltage for all inputs			2.0			Volts
V_{IL}	Input LOW Level	Guaranteed input logical LOW voltage for all inputs		Military			0.7	Volts
				Commercial			0.8	
V_I	Input Clamp Voltage	V_{CC} = MIN., I_{IN} = −18mA					−1.5	Volts
I_{IL}	Input LOW Current	V_{CC} = MAX. V_{IN} = 0.5V	Clock, $\overline{OE}$				−0.36	mA
			A_0, A_1, A_2, A_3				−0.36	
			B_0, B_1, B_2, B_3				−0.36	
			D_0, D_1, D_2, D_3				−0.72	
			I_0, I_1, I_2, I_6, I_8				−0.36	
			I_3, I_4, I_5, I_7				−0.72	
			$RAM_{0,3}, Q_{0,3}$ (Note 4)				−0.8	
			C_n				−3.6	
I_{IH}	Input HIGH Current	V_{CC} = MAX. V_{IN} = 2.7V	Clock, $\overline{OE}$				20	μA
			A_0, A_1, A_2, A_3				20	
			B_0, B_1, B_2, B_3				20	
			D_0, D_1, D_2, D_3				40	
			I_0, I_1, I_2, I_6, I_8				20	
			I_3, I_4, I_5, I_7				40	
			$RAM_{0,3}, Q_{0,3}$ (Note 4)				100	
			C_n				200	
I_I	Input HIGH Current	V_{CC} = MAX., V_{IN} = 5.5V					1.0	mA
I_{OZH} I_{OZL}	Off State (High Impedance) Output Current	V_{CC} = MAX.	Y_0, Y_1, Y_2, Y_3	V_O = 2.4V			50	μA
				V_O = 0.5V			−50	
			$RAM_{0,3}, Q_{0,3}$	V_O = 2.4V (Note 4)			100	
				V_O = 0.5V (Note 4)			−800	
I_{OS}	Output Short Circuit Current (Note 3)	V_{CC} = 5.75V V_O = 0.5V	$Y_0, Y_1, Y_2, Y_3, \overline{G}$		−15		−40	mA
			C_{n+4}		−15		−40	
			OVR, $\overline{P}$		−15		−40	
			F_3		−15		−40	
			$RAM_{0,3}, Q_{0,3}$		−15		−40	
I_{CC}	Power Supply Current	V_{CC} = MAX.	Military			185	280	mA
			Commercial			185	280	

Notes: 1. For conditions shown as MIN. or MAX., use the appropriate value specified under Electrical Characteristics for the applicable device type.
2. Typical limits are at V_{CC} = 5.0V, 25°C ambient and maximum loading.
3. Not more than one output should be shorted at a time. Duration of the short circuit test should not exceed one second.
4. These are three-state outputs internally connected to TTL inputs. Input characteristics are measured with I_{678} in a state such that the three-state output is OFF.

GUARANTEED OPERATING CONDITIONS OVER TEMPERATURE AND VOLTAGE

Tables I, II, and III below define the timing requirements of the MC2901 in a system. The MC2901 is guaranteed to function correctly over the operating range when used within the delay and set-up time constraints of these tables for the appropriate device type. The tables are divided into three types of parameters; clock characteristics, combinational delays from inputs to outputs, and set-up and hold time requirements. The latter table defines the time prior to the end of the cycle (i.e., clock LOW-to-HIGH transition) that each input must be stable to guarantee that the correct data is written into one of the internal registers.

TABLE I

CYCLE TIME AND CLOCK CHARACTERISTICS

TIME	MC2901C	MC2901M
Read-Modify-Write Cycle (time from selection of A, B registers to end of cycle)	105ns	120ns
Maximum Clock Frequency to Shift Q Register (50% duty cycle)	9.5MHz	8.3MHz
Minimum Clock LOW Time	30ns	30ns
Minimum Clock HIGH Time	30ns	30ns
Minimum Clock Period	105ns	120ns

TABLE II

MAXIMUM COMBINATIONAL PROPAGATION DELAYS (all in ns, $C_L \leq 15pF$)

From Input \ To Output	MC2901C (0°C to +70°C; 5V ±5%)								MC2901M (−55°C to +125°C; 5V ±10%)							
	Y	F_3	C_{n+4}	$\overline{G}$, $\overline{P}$	F=0 R_L=470	OVR	Shift Outputs RAM_0 RAM_3	Shift Outputs Q_0 Q_3	Y	F_3	C_{n+4}	$\overline{G}$, $\overline{P}$	F=0 R_L=470	OVR	Shift Outputs RAM_0 RAM_3	Shift Outputs Q_0 Q_3
A, B	110	85	80	80	110	75	110	–	120	95	90	90	120	85	120	–
D (arithmetic mode)	100	70	70	70	100	60	95	–	110	80	75	75	110	65	105	–
D (I = X37) (Note 5)	60	50	–	–	60	–	60	–	65	55	–	–	65	–	65	–
C_n	55	35	30	–	50	40	55	–	60	40	30	–	55	45	60	–
I_{012}	85	65	65	65	80	65	80	–	90	70	70	70	85	70	85	–
I_{345}	70	55	60	60	70	60	65	–	75	60	65	65	75	65	70	–
I_{678}	55	–	–	–	–	–	45	45	60	–	–	–	–	–	50	50
$\overline{OE}$ Enable/Disable	40/25	–	–	–	–	–	–	–	40/25	–	–	–	–	–	–	–
A bypassing ALU (I = 2xx)	60	–	–	–	–	–	–	–	65	–	–	–	–	–	–	–
Clock ⚡ (Note 6)	115	85	100	100	110	95	105	60	125	95	110	110	120	105	115	65

SET-UP AND HOLD TIMES (all in ns) (Note 1)

TABLE III

From Input	Notes	MC2901C (0°C to +70°C, 5V ±5%)		MC2901M (−55°C to +125°C, 5V ±10%)	
		Set-Up Time	Hold Time	Set-Up Time	Hold Time
A, B Source	2, 4 3, 5	105 $t_{pw}L + 30$	0	120 $t_{pw}L + 30$	0
B Dest.	2, 4	$t_{pw}L + 15$	0	$t_{pw}L + 15$	0
D (arithmetic mode)		100	0	110	0
D (I = X37) (Note 5)		60	0	65	0
C_n		55	0	60	0
I_{012}		85	0	90	0
I_{345}		70	0	75	0
I_{678}	4	$t_{pw}L + 15$	0	$t_{pw}L + 15$	0
$RAM_{0, 3}$, $Q_{0, 3}$		30	0	30	0

Notes: 1. See Figure 11 and 12.
2. If the B address is used as a source operand, allow for the "A, B source" set-up time; if it is used only for the destination address, use the "B dest." set-up time.
3. Where two numbers are shown, both must be met.
4. "$t_{pw}L$" is the clock LOW time.
5. DV0 is the fastest way to load the RAM from the D inputs. This function is obtained with I = 337.
6. Using Q register as source operand in arithmetic mode. Clock is not normally in critical speed path when Q is not a source.

SET-UP AND HOLD TIMES (minimum cycles from each input)

Set-up and hold times are defined relative to the clock LOW-to-HIGH edge. Inputs must be steady at all times from the set-up time prior to the clock until the hold time after the clock. The set-up times allow sufficient time to perform the correct operation on the correct data so that the correct ALU data can be written into one of the registers.

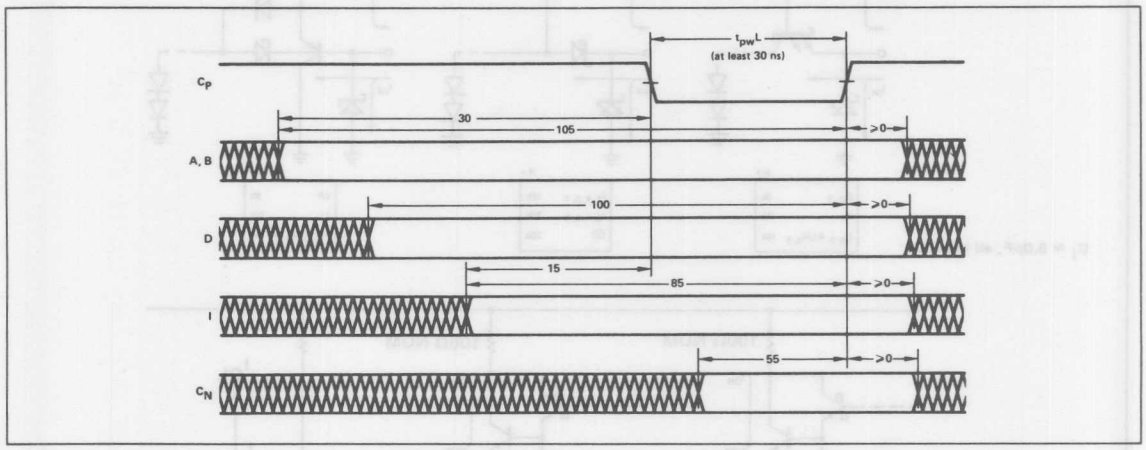
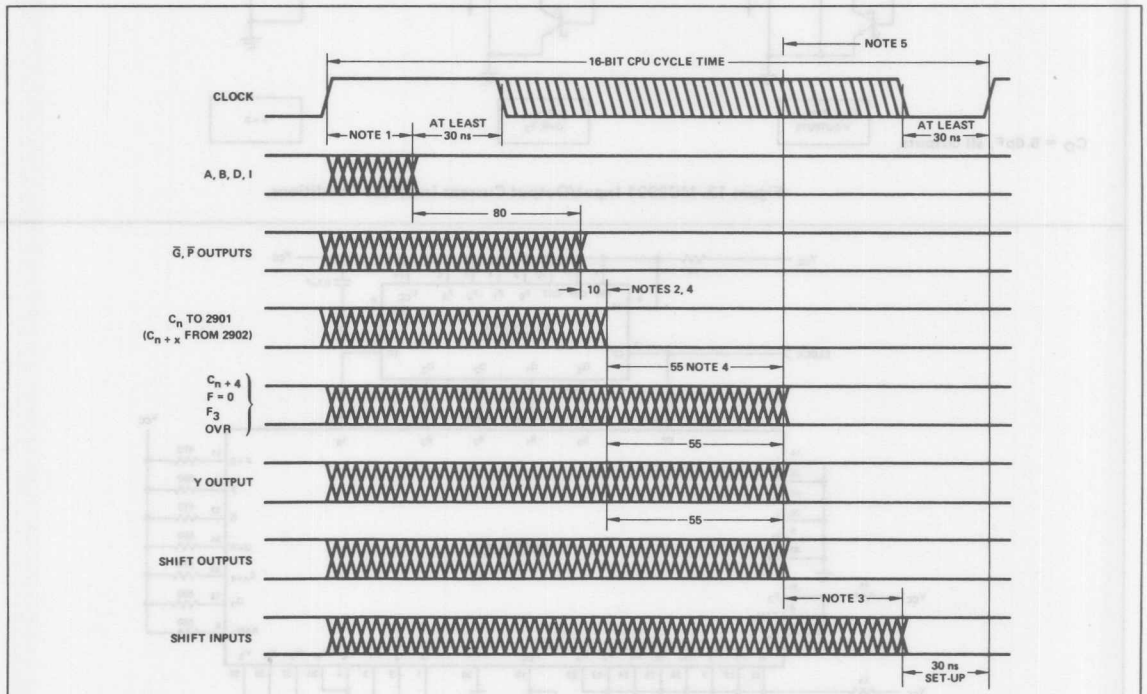


Figure 11. Minimum Cycle Times from Inputs. Numbers Shown are Minimum Data Stable Times for MC2901LC , in ns. See Table III for Detailed Information.



Notes: 1. This delay is the max. t_{pd} of the register containing A, B, D, and I. For the MC2918, use 13 ns.
2. 10ns for look-ahead carry. For ripple carry over 16 bits use 2 x ($C_n \rightarrow C_{n+4}$), or 60ns.
3. This is the delay associated with the multiplexer between the shift outputs and shift inputs on the MC2901s. See Figure 19.
4. Not applicable for logic operations.
5. Clock rising edge may occur here if add and shift do not occur on same cycle.

Figure 12. Switching Waveforms for 16-Bit System Assuming A, B, D and I are all Driven from Registers with the same Propagation Delay, Clocked by the MC2901 Clock.

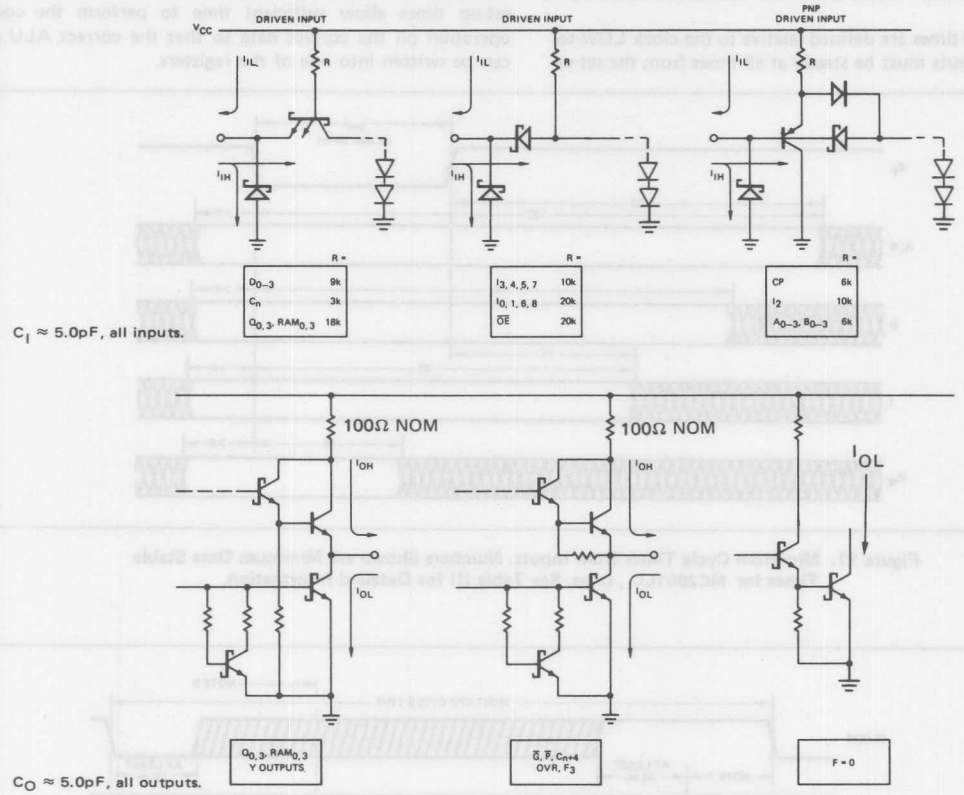


Figure 13. MC2901 Input/Output Current Interface Conditions.

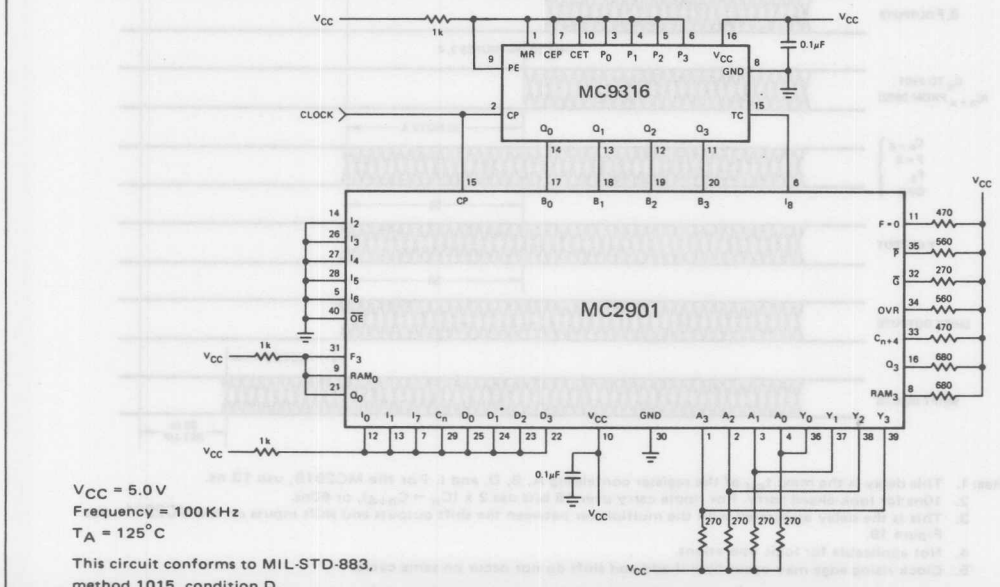


Figure 14. MC2901 Burn-In Circuit.

USING THE MC2901

BASIC SYSTEM ARCHITECTURE

The MC2901 is designed to be used in microprogrammed systems. Figure 15 illustrates such an architecture. The nine instruction lines, the A and B addresses, and the D data inputs normally will all come from registers clocked at the same time as the MC2901. The register inputs come from a ROM or PROM – the "microprogram store" This memory contains sequences of microinstructions, typically 28 to 40 bits wide, which apply the proper control signals to the MC2901's and other circuits to execute the desired operation.

The address lines of the microprogram store are driven from the MC2909 microprogram sequencer. This device has facilities for storing an address, incrementing an address, jumping to any address, and linking subroutines. The MC2909 is controlled by some of the bits coming from the microprogram store. Essentially these bits are the "next instruction" control.

Note that with the microprogram register in-between the microprogram memory store and the MC2901's, an instruction accessed on one cycle is executed on the next cycle. As one instruction is executed, the next instruction is being read from microprogram memory. In this configuration, system speed is improved because the execution time in the MC2901's occurs in parallel with the access time of the microprogram store. Without the "pipeline register", these two functions must occur serially.

EXPANSION OF THE MC2901

The MC2901 is a four-bit CPU slice. Any number of MC2901's can be interconnected to form CPU's of 12, 16, 24, 36 or more bits, in four-bit increments. Figure 16 illustrates the interconnection of three MC2901's to form a 12-bit CPU, using ripple carry. Figure 17 illustrates a 16-bit CPU using carry lookahead, and Figure 18 is the general carry lookahead scheme for long words.

With the exception of the carry interconnection, all expansion schemes are the same. Refer to Figure 16. The Q_3 and RAM_3 pins are bidirectional left/right shift lines at the MSB of the device. For all devices except the most significant, these lines are connected to the Q_0 and RAM_0 pins of the adjacent more significant device. These connections allow the Q-registers of all MC2901's to be shifted left or right as a contiguous n-bit register, and also allow the ALU output data to be shifted left or right as a contiguous n-bit word prior to storage in the RAM. At the LSB and MSB of the CPU, the shift pins should be connected to three-state multiplexers which can be controlled by the microcode to select the appropriate input signals to the shift inputs. (See Figure 19)

The open collector F = 0 outputs of all the MC2901's are connected together and to a pull-up resistor. This line will go HIGH if and only if the output of the ALU contains all zeroes. Most systems will use this line as the Z (zero) bit of the processor status word.

The overflow and F_3 pins are generally used only at the most significant end of the array, and are meaningful only when two's complement signed arithmetic is used. The overflow pin is the Exclusive-OR of the carry-in and carry-out of the sign bit (MSB). It will go HIGH when the result of an arithmetic

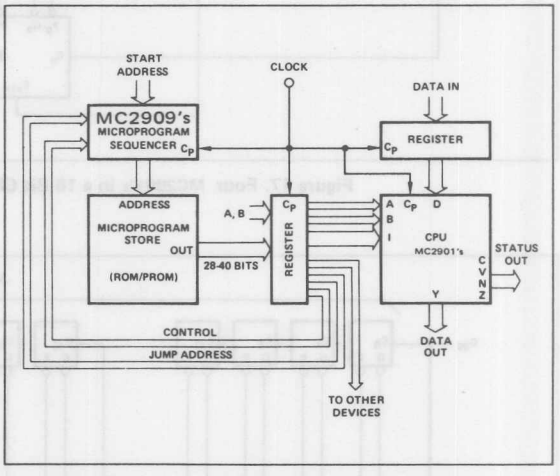


Figure15. Microprogrammed Architecture Around MC2901's.

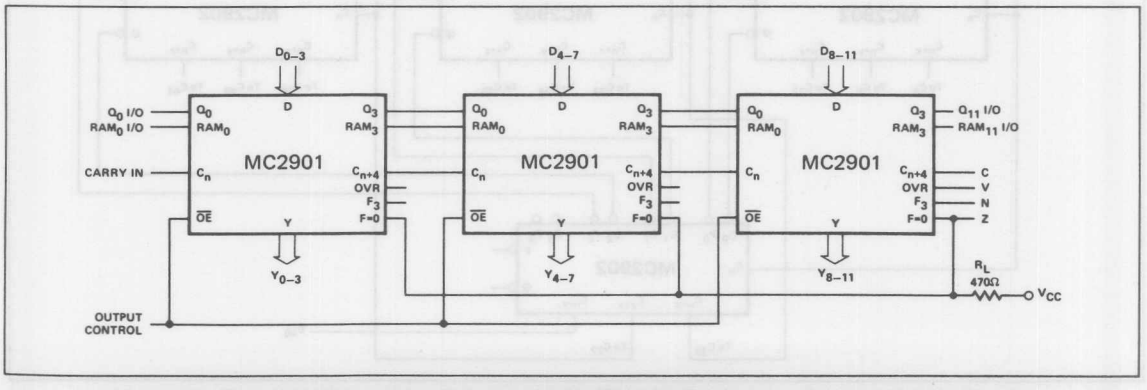


Figure 16. Three MC2901's used to Construct 12-Bit CPU with Ripple Carry. Corresponding A, B, and I Pins on all Devices are Connected Together.

operation is a number requiring more bits than are available, causing the sign bit to be erroneous. This is the overflow (V) bit of the processor status word. The F_3 pin is the MSB of the ALU output. It is the sign of the result in two's complement notation, and should be used as the Negative (N) bit of the processor status word.

The carry-out from the most significant MC2901 (C_{n+4} pin) is the carry-out from the array, and is used as the carry (C) bit of the processor status word.

Carry interconnections between devices may use either ripple carry or carry lookahead. For ripple carry, the carry-out (C_{n+4}) of each device is connected to the carry-in (C_n) of the next more significant device. Carry lookahead uses the MC2902 lookahead carry generator. The scheme is identical to that used with the 74181/74182. Figures 17 and 18 illustrate single and multiple level lookahead.

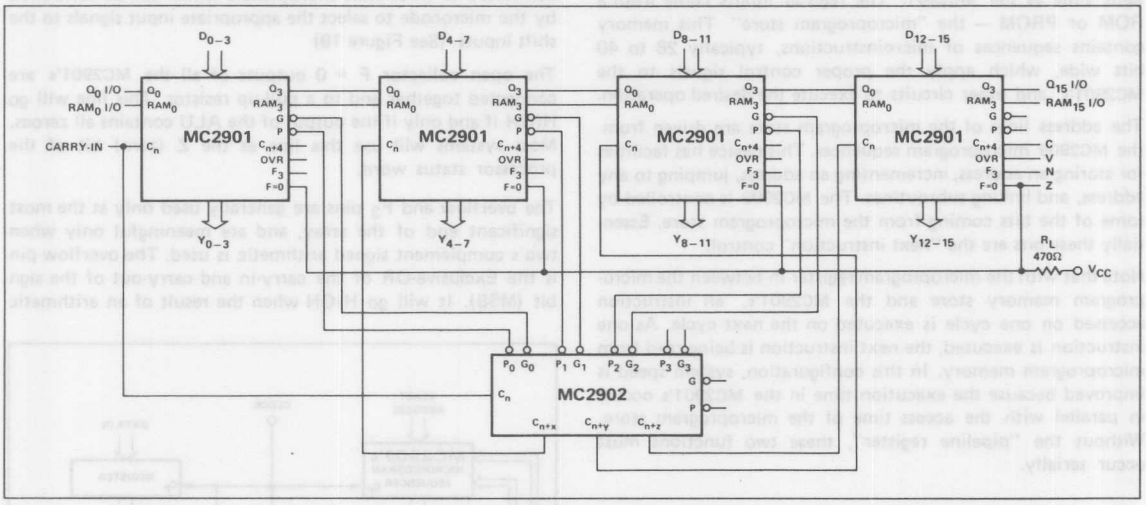


Figure 17. Four MC2901's in a 16-Bit CPU using the MC2902 for Carry Lookahead.

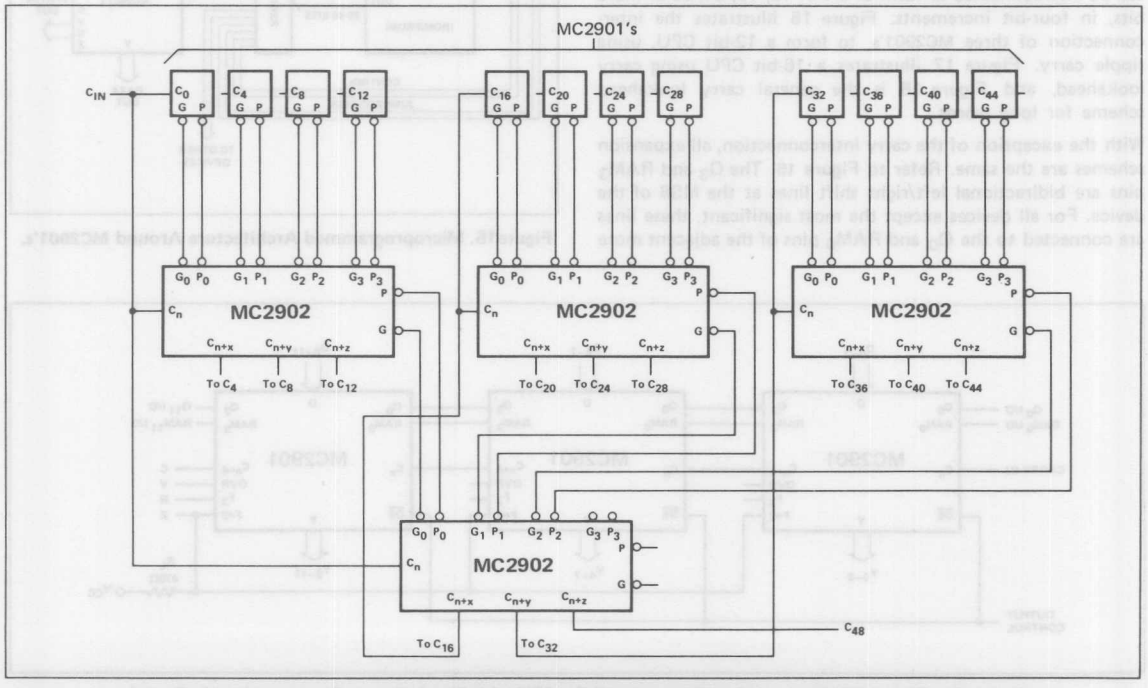


Figure 18. Carry Lookahead Scheme for 48-Bit CPU using 12 MC2901's. The Carry-Out Flag (C_{48}) Should be Taken from the Lower MC2902 Rather than the Right-Most MC2901 for Higher Speed.

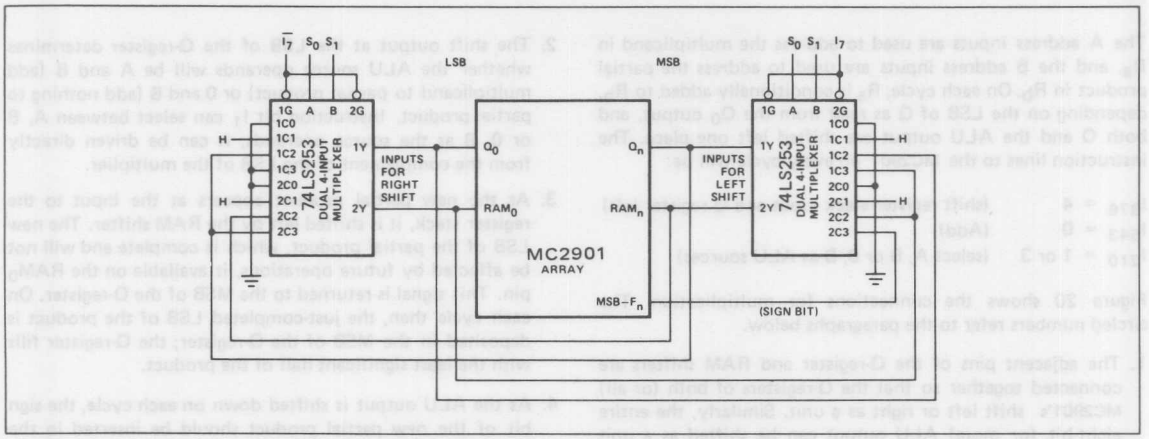


Figure 19. Three-State Multiplexers Used on Shift I/O Lines.

SHIFT I/O LINES AT THE END OF THE ARRAY

The Q-register and RAM left/right shift data transfers occur between devices over bidirectional lines. At the ends of the array, three-state multiplexers are used to select what the new inputs to the registers should be during shifting. Figure 19 shows two 74LS253 dual four-input multiplexers connected to provide four shift modes. Instruction bit I_7 (from the MC 2901) is used to select whether the left-shift multiplexer or the right-shift multiplexer is active. The four shift modes in this example are:

Zero A LOW is shifted into the MSB of the RAM on a down shift. If the Q-register is also shifted, then a LOW is deposited in the Q-register MSB. If the RAM or both registers are shifted up, LOWs are placed in the LSBs.

One Same as zero, but a HIGH level is deposited in the LSB or MSB.

Rotate A single precision rotate. The RAM MSB shifts into the LSB on a right shift and the LSB shifts into the MSB on a left shift. The Q-register, if shifted, will rotate in the same manner.

Arithmetic A double-length Arithmetic Shift if Q is also shifted. On an up shift a zero is loaded into the Q-register LSB and the Q-register MSB is loaded into the RAM LSB. On a down shift, the RAM LSB is loaded into the Q-register MSB and the ALU output MSB (F_n, the sign bit) is loaded into the RAM MSB. (This same bit will also be in the next less significant RAM bit.)

Code			Source of New Data				Shift	Type
I_7	S_1	S_0	Q_0	Q_n	RAM_0	RAM_n		
H	L	L	0	Q_{n-1}	0	F_{n-1}	Up	Zero
H	L	H	1	Q_{n-1}	1	F_{n-1}	(Right)	One
H	H	L	Q_n	Q_{n-1}	F_n	F_{n-1}		Rotate
H	H	H	0	Q_{n-1}	Q_n	F_{n-1}		Arithmetic
L	L	L	Q_1	0	F_1	0	Down	Zero
L	L	H	Q_1	1	F_1	1	(Left)	One
L	H	L	Q_1	Q_0	F_1	F_0		Rotate
L	H	H	Q_1	F_0	F_1	$RAM_n = RAM_{n-1} = F_n$		Arithmetic

HARDWARE MULTIPLICATION

Figure 20 illustrates the interconnections for a hardware multiplication using the MC2901. The system shown uses two devices for 8 x 8 multiplication, but the expansion to more bits is simple – the significant connections are at the LSB and MSB only.

The basic technique used is the "add and shift" algorithm. One clock cycle is required for each bit of the multiplier. On each cycle, the LSB of the multiplier is examined; if it is a "1", then the multiplicand is added to the partial product to generate a new partial product. The partial product is then shifted one place toward the LSB, and the multiplier is also shifted one place toward the LSB. The old LSB of the multiplier is discarded. The cycle is then repeated on the new LSB of the multiplier available at Q_0.

The multiplier is in the MC2901 Q-register. The multiplicand is in one of the registers in the register stack, R_a. The product will be developed in another of the registers in the stack, R_b.

The A address inputs are used to address the multiplicand in R_a, and the B address inputs are used to address the partial product in R_b. On each cycle, R_a is conditionally added to R_b, depending on the LSB of Q as read from the Q_0 output, and both Q and the ALU output are shifted left one place. The instruction lines to the MC2901 on every cycle will be:

I_{876} = 4 (shift register stack input and Q register left)
I_{543} = 0 (Add)
I_{210} = 1 or 3 (select A, B or 0, B as ALU sources)

Figure 20 shows the connections for multiplication. The circled numbers refer to the paragraphs below.

1. The adjacent pins of the Q-register and RAM shifters are connected together so that the Q-registers of both (or all) MC2901's shift left or right as a unit. Similarly, the entire eight-bit (or more) ALU output can be shifted as a unit prior to storage in the register stack.

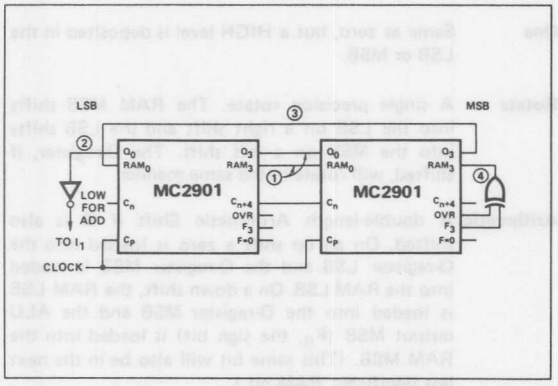


Figure 20. Interconnection for Dedicated Multiplication (8 by 8 Bit) (Corresponding A, B and I Connected Together).

2. The shift output at the LSB of the Q-register determines whether the ALU source operands will be A and B (add multiplicand to partial product) or 0 and B (add nothing to partial product. Instruction bit I_1 can select between A, B or 0, B as the source operands; it can be driven directly from the complement of the LSB of the multiplier.

3. As the new partial product appears at the input to the register stack, it is shifted left by the RAM shifter. The new LSB of the partial product, which is complete and will not be affected by future operations, is available on the RAM_0 pin. This signal is returned to the MSB of the Q-register. On each cycle then, the just-completed LSB of the product is deposited in the MSB of the Q-register; the Q-register fills with the least significant half of the product.

4. As the ALU output is shifted down on each cycle, the sign bit of the new partial product should be inserted in the RAM MSB shift input. The F_3 flag will be the correct sign of the partial product unless overflow has occurred. If overflow occurs during an addition or subtraction, the OVR flag will go HIGH and F_3 is not the sign of the result. The sign of the result must then be the complement of F_3. The correct sign bit to shift into the MSB of the partial product is therefore F3 ⊕ OVR; that is, F_3 if overflow has not occurred and $\overline{F_3}$ if overflow has occurred. On the last cycle, when the MSB of the multiplier is examined, a conditional subtraction rather than addition should be performed, because the sign bit of the multiplier carries negative rather than positive arithmetic weight

$$(Y = -Y_i 2^i + Y_{i-1} 2^{i-1} + \ldots + Y_0 2^0).$$

This scheme will produce a correct two's complement product for all multiplicands and multipliers in two's complement notation.

Figure 21 is a table showing the input states of the MC2901's for each step of a signed, two's complement multiplication.

Initial Register States

R	
0	Multiplier
1	Multiplicand
2	X
3	X

MC2901 Microcode

Program 2's Comp. Multiply

Final Register States

R	
0	Multiplier
1	Multiplicand
2	LSH Product
3	MSH Product

S, F →	D	Description	Repeat	Pin States (Octal)										Jump	
				A	B	I_{876}	I_{543}	I_{210}	C_n	Q_0	Q_3	RAM_0	RAM_3	To	If
O ∨ A	Q	Move Multiplier to Q	–	0	X	0	3	4	X	X	X	X	X		
O ∧ B	B	Clear R_3	–	X	3	2	4	3	X	X	X	X	X		
(O+B)/2 (A+B)/2	B	Cond. Add & Shift	n–1	1	3	4	0	1 or 3 $I_1 = \overline{Q_0}$ LO	0	–	RAM_0	–	$F_3 \forall OVR$		
(B–O)/2 (B–A)/2	B	Cond. Subt. & Shift	–	1	3	4	1	1 or 3 $I_1 = \overline{Q_0}$ LO	1	–	RAM_0	–	$F_3 \forall OVR$		
O ∨ Q	B	Move LSH Prod. to R_2	–	X	2	2	3	2	X	X	X	X	X		

X = Don't Care S = Source F = Function D = Destination

EXAMPLES OF SOME OTHER OPERATIONS

1. **Byte Swapping**

   Occasionally the two halves of a 16-bit word must be swapped. D_{0-7} is interchanged with D_{8-15}. The quickest way to perform this operation is to rotate the word in RAM, shifting two bits at a time. Only four shift cycles are required. The same register is selected on both the A and B ports; the two are added together with carry-in connected to carry-out, producing a right shift of one place; then the ALU is shifted right one more place prior to storage.

   Byte Swap of R_0

   $A = B = 0$ $I = 701$ $RAM_0 = RAM_{15}$ $C_{IN} = C_{OUT}$

   Repeat 4 times

2. **Instruction Fetch Cycle**

   Execution of a macroinstruction generally begins with an instruction fetch cycle. The current contents of the PC (in one of the registers) is the address of the macroinstruction to be fetched, and must be read out to the memory address register. Then the PC is incremented to point to the next macroinstruction. The macroinstruction obtained from memory is then loaded into the MC2909 microprogram sequencer to cause a jump to the microcode for executing the instruction.

   The PC can be read out and incremented in one cycle by using the MC2901 destination code 2, and addressing the PC with both the A and B addresses. The current value of PC will appear on the Y outputs, and PC+1 will be returned to the register. If the PC is in register 15, then:

   A = B = 15, I = 203, Carry-in = 1

   The PC will be on the Y outputs via the RAM A-port. On the clock LOW-to-HIGH transition, the program counter is incremented and the value on the Y outputs is loaded into the memory address register. During the following cycle, the memory is read and, on the next clock LOW-to-HIGH transition the instruction from the memory is dropped into the MC2909 instruction register. The fetch operation requires only two microcycles.

MC2909
MICROPROGRAM SEQUENCER

DISTINCTIVE CHARACTERISTICS

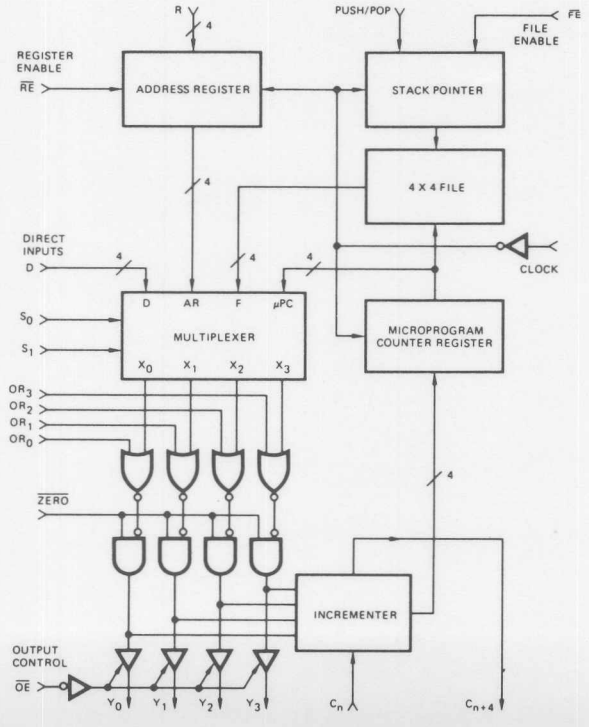
- 4-bit slice cascadable to any number of microwords
- Internal address register
- Branch input for N-way branches
- Cascadable 4-bit microprogram counter
- 4 x 4 file with stack pointer and push pop control for nesting microsubroutines.
- Zero input for returning to the zero microcode word
- Individual OR input for each bit for branching to higher microinstructions
- Three-state outputs
- All internal registers change state on the LOW-to-HIGH transition of the clock

GENERAL DESCRIPTION

The MC2909 is a four-bit wide address controller intended for sequencing through a series of microinstructions contained in a ROM or PROM. Two MC2901's may be interconnected to generate an eight-bit address (256 words), and three may be used to generate a twelve-bit address (4K words).

The MC2909 can select an address from any of four sources. They are: 1) a set of external direct inputs (D); 2) external data from the R inputs, stored in an internal register; 3) a four-word deep push/pop stack; or 4) a program counter register (which usually contains the last address plus one). The push/pop stack includes certain control lines so that it can efficiently execute nested subroutine linkages. Each of the four outputs can be OR'ed with an external input for conditional skip or branch instructions, and a separate line forces the outputs to all zeroes. The outputs are three-state.

MICROPROGRAM SEQUENCER BLOCK DIAGRAM



ARCHITECTURE OF THE MC2909

The MC2909 is a bipolar microprogram sequencer intended for use in high-speed microprocessor applications. The device is a cascadable 4-bit slice such that two devices allow addressing of up to 256-words of microprogram and three devices allow addressing of up to 4K words of microprogram. A detailed logic diagram is shown in Figure 2.

The device contains a four-input multiplexer that is used to select either the address register, direct inputs, microprogram counter, or file as the source of the next microinstruction address. This multiplexer is controlled by the S_0 and S_1 inputs.

The address register consists of four D-type, edge-triggered flip-flops with a common clock enable. When the address register enable is LOW, new data is entered into the register on the clock LOW-to-HIGH transition. The address register is available at the multiplexer as a source for the next microinstruction address. The direct input is a four-bit field of inputs to the multiplexer and can be selected as the next microinstruction address. This allows an N-way branch where N is any word in the microcode.

The MC2909 contains a microprogram counter (μPC) that is composed of a 4-bit incrementer followed by a 4-bit register. The incrementer has carry-in (C_n) and carry-out (C_{n+4}) such that cascading to larger word lengths is straight-forward. The μPC can be used in either of two ways. When the least significant carry-in to the incrementer is HIGH, the microprogram register is loaded on the next clock cycle with the current Y output word plus one (Y+1 → μPC.) Thus sequential microinstructions can be executed. If this least significant C_n is LOW, the incrementer passes the Y output word unmodified and the microprogram register is loaded with the same Y word on the next clock cycle (Y → μPC). Thus, the same microinstruction can be executed any number of times by using the least significant C_n as the control.

The last source available at the multiplexer input is the 4 x 4 file (stack). The file is used to provide return address linkage when executing microsubroutines. The file contains a built-in stack pointer (SP) which always points to the last file word written. This allows stack reference operations (looping) to be performed without a push or pop.

The stack pointer operates as an up/down counter with separate push/pop and file enable inputs. When the file enable input is LOW and the push/pop input is HIGH, the PUSH operation is enabled. This causes the stack pointer to increment and the file to be written with the required return linkage – the next microinstruction address following the subroutine jump which initiated the PUSH.

If the file enable input is LOW and the push/pop control is LOW, a POP operation occurs. This implies the usage of the return linkage during this cycle and thus a return from subroutine. The next LOW-to-HIGH clock transition causes the stack pointer to decrement. If the file enable is HIGH, no action is taken by the stack pointer regardless of any other input.

The stack pointer linkage is such that any combination of pushes, pops or stack references can be achieved. One microinstruction subroutines can be performed. Since the stack is 4 words deep, up to four microsubroutines can be nested.

The ZERO input is used to force the four outputs to the binary zero state. When the ZERO input is LOW, all Y outputs are LOW regardless of any other inputs (except $\overline{OE}$). Each Y output bit also has a separate OR input such that a conditional logic one can be forced at each Y output. This allows jumping to different microinstructions on programmed conditions.

The MC2909 features three-state Y outputs. These can be particularly useful in military designs requiring external Ground Support Equipment (GSE) to provide automatic checkout of the microprocessor. The internal control can be placed in the high-impedance state, and preprogrammed sequences of microinstructions can be executed via external access to the control ROM/PROM.

PHYSICAL DIMENSIONS
28-Pin Package

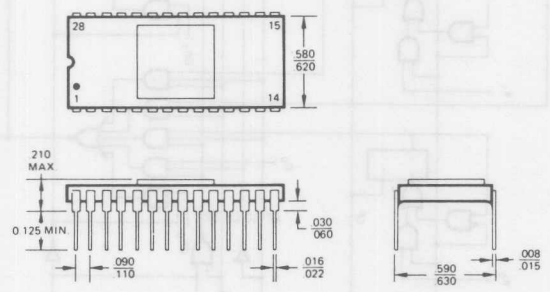


Figure 1.

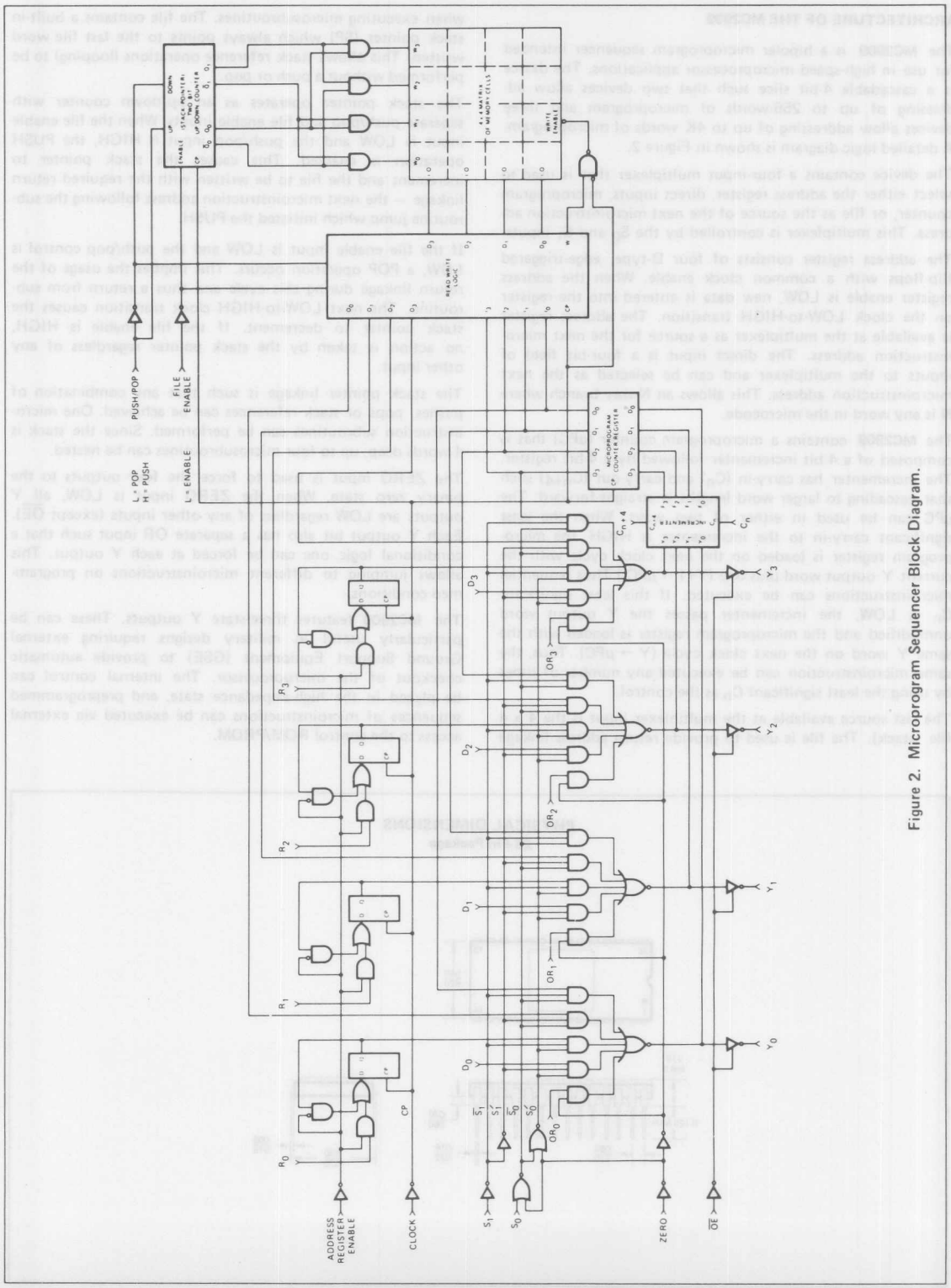


Figure 2. Microprogram Sequencer Block Diagram.

DEFINITION OF TERMS

A set of symbols is used in this data sheet to represent various internal and external registers and signals used with the MC2909. Since its principle application is as a controller for a microprogram store, it is necessary to define some signals associated with the microcode itself. Figure 3 illustrates the basic interconnection of MC2909, memory, and microinstruction register. The definitions here apply to this architecture.

Inputs to MC2909

S_1, S_0	Control lines for address source selection
$\overline{FE}$, PUP	Control lines for push/pop stack
$\overline{RE}$	Enable line for internal address register
OR_i	Logic OR inputs on each address output line
$\overline{ZERO}$	Logic AND input on the output lines
$\overline{OE}$	Output Enable. When $\overline{OE}$ is HIGH, the Y outputs are OFF (high impedance)
C_n	Carry-in to the incrementer
R_i	Inputs to the internal address register
D_i	Direct inputs to the multiplexer
CP	Clock input to the AR and μPC register and Push-Pop stack

Outputs from the MC2909

Y_i	Address outputs from MC2909. (Address inputs to control memory.)
C_{n+4}	Carry out from the incrementer

Internal Signals

μPC	Contents of the microprogram counter
AR	Contents of the address register
STK0-STK3	Contents of the push/pop stack. By definition, the word in the four-by-four file, addressed by the stack pointer is STK0. Conceptually data is pushed into the stack at STK0; a subsequent push moves STK0 to STK1; a pop implies STK3 → STK2 → STK1 → STK0. Physically, only the stack pointer changes when a push or pop is performed. The data does not move. I/O occurs at STK0.
SP	Contents of the stack pointer

External to the MC2909

A	Address to the control memory
I(A)	Instruction in control memory at address A
μWR	Contents of the microword register (at output of control memory). The microword register contains the instruction currently being executed.
T_n	Time period (cycle) n

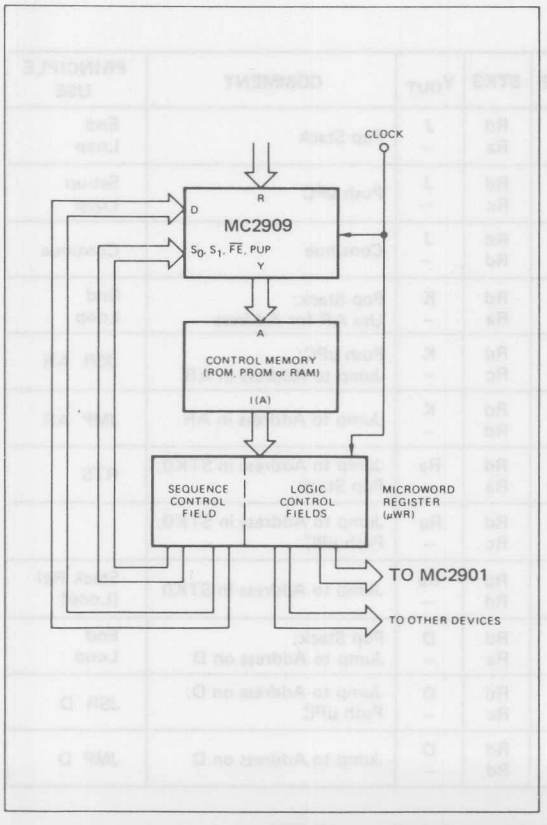
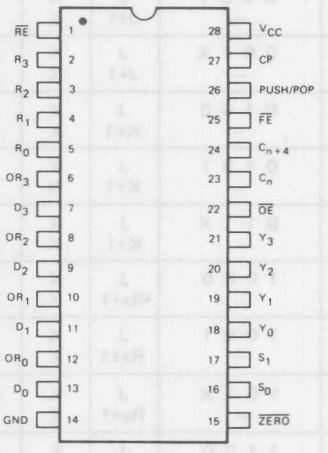


Figure 3. Microprogram Sequencer Control.

CONNECTION DIAGRAM
Top View



Note: Pin 1 is marked for orientation.

Figure 4.

OPERATION OF THE MC2909

Figure 5 lists the select codes for the multiplexer. The two bits applied from the microword register (and additional combinational logic for branching) determine which data source contains the address for the next microinstruction. The contents of the selected source will appear on the Y outputs. Figure 5 also shows the truth table for the output control and for the control of the push/pop stack. Figure 6 shows in detail the effect of S_0, S_1, $\overline{FE}$ and PUP on the MC2909. These four signals define what address appears on the Y outputs and what the state of all the internal registers will be following the clock LOW-to-HIGH edge. In this illustration, the microprogram counter is assumed to contain initially some word J, the address register some word K, and the four words in the push/pop stack contain R_a through R_d.

Address Selection

OCTAL	S_1	S_0	SOURCE FOR Y OUTPUTS	SYMBOL
0	L	L	Microprogram Counter	μPC
1	L	H	Address register	AR
2	H	L	Push-Pop stack	STK0
3	H	H	Direct inputs	D_i

Output Control

OR_i	$\overline{ZERO}$	$\overline{OE}$	Y_i
X	X	H	Z
X	L	L	L
H	H	L	H
L	H	L	Source selected by S_0 S_1

Z = High Impedance

Synchronous Stack Control

$\overline{FE}$	PUP	PUSH-POP STACK CHANGE
H	X	No change
L	H	Increment stack pointer, then push current PC onto STK0
L	L	Pop stack (decrement stack pointer)

Figure 5.

CYCLE	S_1, S_0, $\overline{FE}$, PUP	μPC	AR	STK0	STK1	STK2	STK3	Y_{OUT}	COMMENT	PRINCIPLE USE
N	0 0 0 0	J	K	Ra	Rb	Rc	Rd	J	Pop Stack	End Loop
N+1	–	J+1	K	Rb	Rc	Rd	Ra	–		
N	0 0 0 1	J	K	Ra	Rb	Rc	Rd	J	Push μPC	Set-up Loop
N+1	–	J+1	K	J	Ra	Rb	Rc	–		
N	0 0 1 X	J	K	Ra	Rb	Rc	Rd	J	Continue	Continue
N+1	–	J+1	K	Ra	Rb	Rc	Rd	–		
N	0 1 0 0	J	K	Ra	Rb	Rc	Rd	K	Pop Stack; Use AR for Address	End Loop
N+1	–	K+1	K	Rb	Rc	Rd	Ra	–		
N	0 1 0 1	J	K	Ra	Rb	Rc	Rd	K	Push μPC; Jump to Address in AR	JSR AR
N+1	–	K+1	K	J	Ra	Rb	Rc	–		
N	0 1 1 X	J	K	Ra	Rb	Rc	Rd	K	Jump to Address in AR	JMP AR
N+1	–	K+1	K	Ra	Rb	Rc	Rd	–		
N	1 0 0 0	J	K	Ra	Rb	Rc	Rd	Ra	Jump to Address in STK0; Pop Stack	RTS
N+1	–	Ra+1	K	Rb	Rc	Rd	Ra	–		
N	1 0 0 1	J	K	Ra	Rb	Rc	Rd	Ra	Jump to Address in STK0; Push μPC	
N+1	–	Ra+1	K	J	Ra	Rb	Rc	–		
N	1 0 1 X	J	K	Ra	Rb	Rc	Rd	Ra	Jump to Address in STK0	Stack Ref (Loop)
N+1	–	Ra+1	K	Ra	Rb	Rc	Rd	–		
N	1 1 0 0	J	K	Ra	Rb	Rc	Rd	D	Pop Stack; Jump to Address on D	End Loop
N+1	–	D+1	K	Rb	Rc	Rd	Ra	–		
N	1 1 0 1	J	K	Ra	Rb	Rc	Rd	D	Jump to Address on D; Push μPC	JSR D
N+1	–	D+1	K	J	Ra	Rb	Rc	–		
N	1 1 1 X	J	K	Ra	Rb	Rc	Rd	D	Jump to Address on D	JMP D
N+1	–	D+1	K	Ra	Rb	Rc	Rd	–		

X = Don't Care, 0 = LOW, 1 = HIGH, Assume C_n = HIGH

Figure 6. Output and Internal Next-Cycle Register States for MC2909.

Figure 7 illustrates the execution of a subroutine using the MC2909. The configuration of Figure 3 is assumed. The instruction being executed at any given time is the one contained in the microword register (μWR). The contents of the μWR also controls (indirectly, perhaps) the four signals S_0, S_1, $\overline{FE}$, and PUP. The starting address of the subroutine is applied to the D inputs of the MC2909 at the appropriate time.

In the columns on the left is the sequence of microinstructions to be executed. At address J+2, the sequence control portion of the microinstruction contains the command "Jump to subroutine at A". At the time T_2, this instruction is in the μWR, and the MC2909 inputs are set-up to execute the jump and save the return address. The subroutine address A is applied to the D inputs from the μWR and appears on the Y outputs. The first instruction of the subroutine, I(A), is accessed and is at the inputs of the μWR. On the next clock transition, I(A) is loaded into the μWR for execution, and the return address J+3 is pushed onto the stack. The return instruction is executed at T_5. Figure 8 is a similar timing chart showing one subroutine linking to a second, the latter consisting of only one microinstruction.

CONTROL MEMORY

Execute Cycle	Microprogram Address	Microprogram Instruction
	J−1	–
T_0	J	–
T_1	J+1	–
T_2	J+2	JSR A
T_6	J+3	–
T_7	J+4	–
	–	–
	–	–
	–	–
	–	–
	–	–
T_3	A	I(A)
T_4	A+1	–
T_5	A+2	RTS
	–	–
	–	–
	–	–
	–	–
	–	–
	–	–

Execute Cycle		T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
Clock											
Signals											
MC2909 Inputs (from μWR)	S_1, S_0	0	0	3	0	0	2	0	0		
	$\overline{FE}$	H	H	L	H	H	L	H	H		
	PUP	X	X	H	X	X	L	X	X		
	D	X	X	A	X	X	X	X	X		
Internal Registers	μPC	J+1	J+2	J+3	A+1	A+2	A+3	J+4	J+5		
	STK0	–	–	–	J+3	J+3	J+3	–	–		
	STK1	–	–	–	–	–	–	–	–		
	STK2	–	–	–	–	–	–	–	–		
	STK3	–	–	–	–	–	–	–	–		
MC2909 Output	Y	J+1	J+2	A	A+1	A+2	J+3	J+4	J+5		
ROM Output	(Y)	I(J+1)	**JSR A**	I(A)	I(A+1)	**RTS**	I(J+3)	I(J+4)	I(J+5)		
Contents of μWR (Instrution being executed)	μWR	I(J)	I(J+1)	**JSR A**	I(A)	I(A+1)	**RTS**	I(J+3)	I(J+4)		

Figure 7. Subroutine Execution. C_n = HIGH

CONTROL MEMORY

Execute Cycle	Microprogram Address	Microprogram Instruction
	J−1	–
T_0	J	–
T_1	J+1	–
T_2	J+2	JSR A
T_9	J+3	–
	–	–
	–	–
	–	–
	–	–
T_3	A	–
T_4	A+1	–
T_5	A+2	JSR B
T_7	A+3	–
T_8	A+4	RTS
	–	–
	–	–
	–	–
	–	–
T_6	B	RTS
	–	–

Execute Cycle		T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
Clock											
Signals											
MC2909 Inputs (from μWR)	S_1, S_0	0	0	3	0	0	3	2	0	2	0
	$\overline{FE}$	H	H	L	H	H	L	L	H	L	H
	PUP	X	X	H	X	X	H	L	X	L	X
	D	X	X	A	X	X	B	X	X	X	X
Internal Registers	μPC	J+1	J+2	J+3	A+1	A+2	A+3	B+1	A+4	A+5	J+4
	STK0	–	–	–	J+3	J+3	J+3	A+3	J+3	J+3	–
	STK1	–	–	–	–	–	–	J+3	–	–	–
	STK2	–	–	–	–	–	–	–	–	–	–
	STK3	–	–	–	–	–	–	–	–	–	–
MC2909 Output	Y	J+1	J+2	A	A+1	A+2	B	A+3	A+4	J+3	J+4
ROM Output	(Y)	I(J+1)	**JSR A**	I(A)	I(A+1)	**JSR B**	**RTS**	I(A+3)	**RTS**	I(J+3)	I(J+4)
Contents of μWR (Instrution being executed)	μWR	I(J)	I(J+1)	**JSR A**	I(A)	I(A+1)	**JSR B**	**RTS**	I(A+3)	**RTS**	I(J+3)

Figure 8. Two Nested Subroutines. Routine B is Only One Instruction. C_n = HIGH

MAXIMUM RATINGS (Above which the useful life may be impaired)

Storage Temperature	−65°C to +150°C
Temperature (Ambient) Under Bias	−55°C to +125°C
Supply Voltage to Ground Potential	−0.5 V to +7.0 V
DC Voltage Applied to Outputs for HIGH Output State	−0.5 V to +V_{CC} max.
DC Input Voltage	−0.5 V to +7.0 V
DC Output Current, Into Outputs	30 mA
DC Input Current	−30 mA to +5.0 mA

OPERATING RANGE

P/N	Ambient Temperature	V_{CC}
MC2909LC	0°C to +70°C	4.75 V to 5.25 V
MC2909LM	−55°C to +125°C	4.50 V to 5.50 V

ORDERING INFORMATION

Package Type	Temperature Range	Order Number
Hermetic DIP	0°C to +70°C	MC2909LC
Hermetic DIP	−55°C to +125°C	MC2909LM
Dice	0°C to +70°C	MCC2909C

ELECTRICAL CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (Unless Otherwise Noted)

Parameters	Description	Test Conditions (Note 1)			Min.	Typ. (Note 2)	Max.	Units
V_{OH}	Output HIGH Voltage	V_{CC} = MIN., V_{IN} = V_{IH} or V_{IL}	MIL	I_{OH} = −1.0mA	2.4			Volts
			COM'L	I_{OH} = −2.6mA	2.4			
V_{OL}	Output LOW Voltage	V_{CC} = MIN., V_{IN} = V_{IH} or V_{IL}	I_{OL} = 4.0mA				0.4	Volts
			I_{OL} = 8.0mA				0.45	
			I_{OL} = 12mA (Note 5)				0.5	
V_{IH}	Input HIGH Level	Guaranteed input logical HIGH voltage for all inputs			2.0			Volts
V_{IL}	Input LOW Level	Guaranteed input logical LOW voltage for all inputs		MIL			0.7	Volts
				COM'L			0.8	
V_I	Input Clamp Voltage	V_{CC} = MIN., I_{IN} = −18mA					−1.5	Volts
I_{IL}	Input LOW Current	V_{CC} = MAX., V_{IN} = 0.4V	C_n				−1.08	mA
			Push/Pop, OE				−0.72	
			Others				−0.36	
I_{IH}	Input HIGH Current	V_{CC} = MAX., V_{IN} = 2.7V	C_n				40	μA
			Push/Pop				40	
			Others				20	
I_I	Input HIGH Current	V_{CC} = MAX., V_{IN} = 7.0V	C_n, Push/Pop				0.2	mA
			Others				0.1	
I_{OS}	Output Short Circuit Current (Note 3)	V_{CC} = MAX.			−40		−100	mA
I_{CC}	Power Supply Current	V_{CC} = MAX. (Note 4)				80	130	mA
I_{OZL}	Output OFF Current	V_{CC} = MAX., $\overline{OE}$ = 2.7V	V_{OUT} = 0.4V				−20	μA
I_{OZH}			V_{OUT} = 2.7V				20	

Notes: 1. For conditions shown as MIN. or MAX., use the appropriate value specified under Electrical Characteristics for the applicable device type.
2. Typical limits are at V_{CC} = 5.0V, 25°C ambient and maximum loading.
3. Not more than one output should be shorted at a time. Duration of the short circuit test should not exceed one second.
4. Apply GND to C_n, R_0, R_1, R_2, R_3, OR_0, OR_1, OR_2, OR_3, D_0, D_1, D_2, and D_3. Other inputs open. All outputs open. Measured after a LOW-to-HIGH clock transition.
5. The 12mA output applies only to Y_0, Y_1, Y_2 and Y_3.

SWITCHING CHARACTERISTICS OVER OPERATING RANGE

All parameters are guaranteed worst case over the operating voltage and temperature range for the device type.
(Grade C = 0°C to +70°C, 4.75V to 5.25V; Grade M = −55°C to +125°C, 4.5V to 5.5V)

TABLE I
MINIMUM CLOCK REQUIREMENTS

Minimum Clock LOW Time	50
Minimum Clock HIGH Time	30

TABLE II
MAXIMUM COMBINATORIAL PROPAGATION DELAYS

OUTPUTS / INPUTS	Y_i	C_{n+4}
$\overline{OE}$	25	–
$\overline{ZERO}$	35	45
OR_i	20	32
S_0, S_1	40	50
D_i	20	32
C_n	–	18

TABLE III
MAXIMUM DELAYS FROM CLOCK TO OUTPUTS

FUNCTIONAL PATH	GRADE	CLOCK TO Y_i	CLOCK TO C_{n+4}
Register (S_1 S_0 = LH)	C	48	58
	M	55	65
μ Program Counter (S_1 S_0 = LL)	C	48	58
	M	55	65
File (S_1 S_0 = HL)	C	70	80
	M	80	90

$R_L = 2.0k\Omega$ $C_L = 15pF$

TABLE IV
SET-UP AND HOLD TIME REQUIREMENTS

EXTERNAL INPUTS	t_s	t_h
$\overline{RE}$	20	5.0
R_i	15	0
PUSH/POP	20	5.0
$\overline{FE}$	20	0
C_n	15	0
D_i	20	0
OR_i	20	0
S_0, S_1	40	0
$\overline{ZERO}$	40	0

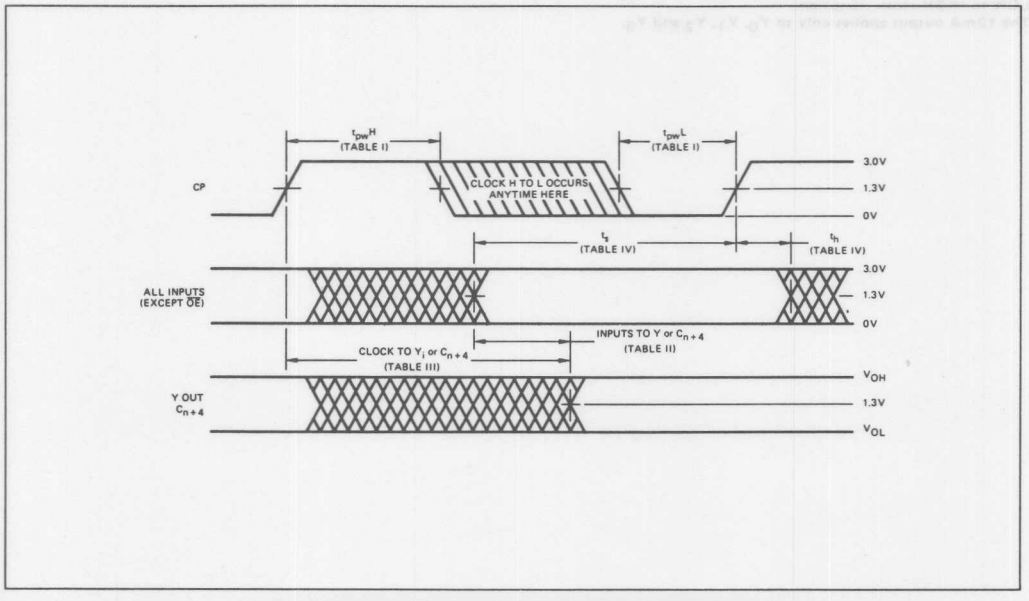


Figure 12. Switching Waveforms. See Tables for Specific Values.

USING THE MC2909

INTRODUCTION

The computer control unit (CCU) is generally the single most complicated subsystem in today's digital computer. A CCU is complicated from the conceptualization, design and implementation viewpoints, because it is the subsystem that controls the internal buses and subsystems of the processor, synchronizes internal and external events and grants or denies permission to external systems. The MC2909 Microprogram Sequencer is an excellent mechanism for simplifying the CCU design task.

COMPUTER ARCHITECTURE

A classical computer architecture is shown in Figure 1. The data bus is commonly used by all of the subsystems in the computer. Information, instructions, address operands, data and sometimes control signals are transmitted down the data bus under control of a microprogram. The microprogram selects the source of the data as well as the destination (s) of the data. In a more complicated system there may be a number of data buses.

The address bus is typically used to select a word in memory for an internal computer function, or to select an input/output port for an external subsystem or peripheral function. Also selected by microprogram command, the source of the data for the address bus may be the program counter, the memory address register, a direct memory address controller, an interface controller etc.

The arithmetic/logic unit (ALU) is actually that portion of the processor that computes. Depending upon the complexity of the ALU, a large number of different arithmetic functions can be accomplished in various number system using different representations of those data. The most common minimum set, however, are the functions (A plus B), (A minus B) and (B

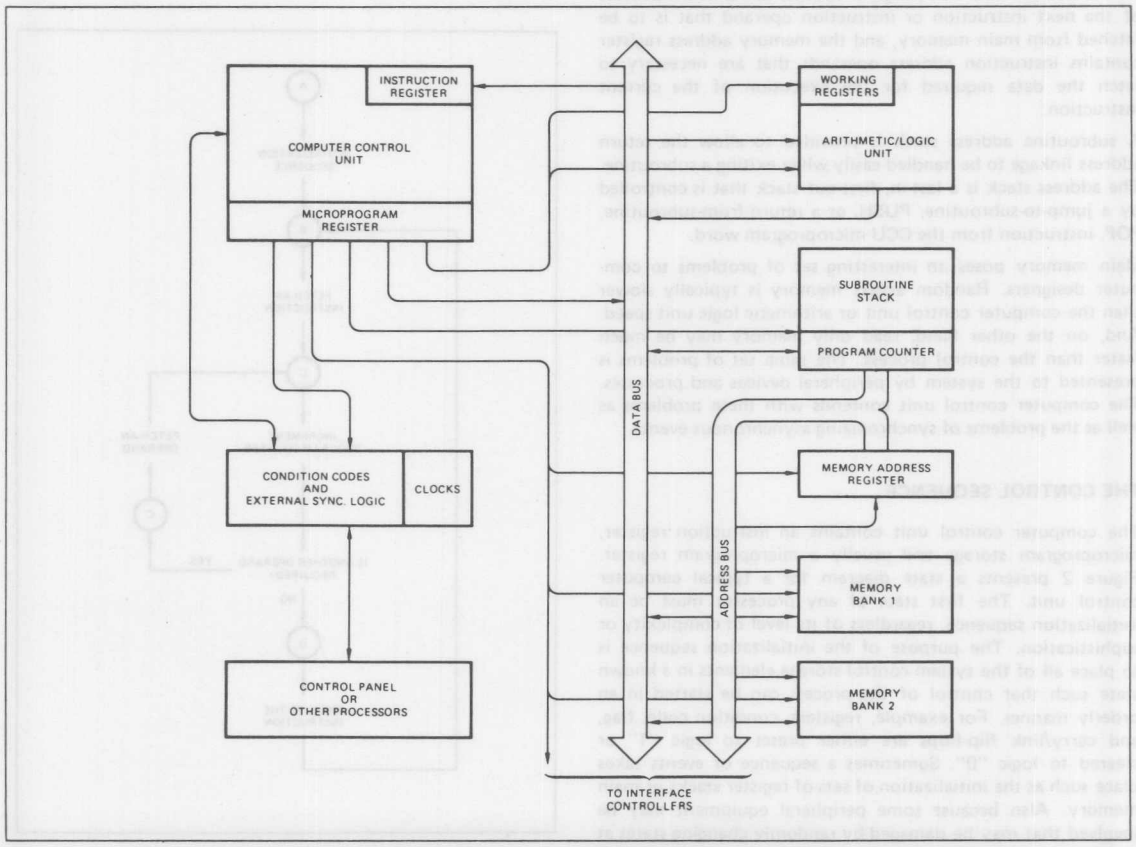


Figure 1. Generalized Computer Architecture.

minus A) performed in fixed point, two's complement binary form; where A and B are the ALU inputs. The logical functions are obtained from the same combinatorial logic array that is used for the arithmetic functions, but it is gated in a different manner. The minimum logical function capability will be (A OR B), (A AND B) and (A EXCLUSIVE-OR B). In addition to these combinatorial logic functions, there are sets of shift and rotate instructions that complete the basic instruction set.

The ALU provides a set of condition codes as a result of the current arithmetic or logical function. These condition codes include such variables as carry out, A = B, the sign bit, result equals zero, etc. The condition codes, along with other computer status information, are stored in a register for later use by the programmer or computer control unit.

Third generation processors also provide for a general-purpose register set that is available to the programmer to be used to hold variables that are used often – passing arguments to subroutines, referencing memory indirectly etc. Depending on the architecture of the machine, the general-purpose registers may be selected directly from the operands in the instruction register, from an address in the microprogram store, or one of the two sources as determined by a bit in the microprogram store.

The program counter and the memory address register are the two main sources of memory word and I/O address select data on the address bus. The program counter contains the address of the next instruction or instruction operand that is to be fetched from main memory, and the memory address register contains instruction address operands that are necessary to fetch the data required for the execution of the current instruction.

A subroutine address stack is provided to allow the return address linkage to be handled easily when exiting a subroutine. The address stack is a last-in, first-out stack that is controlled by a jump-to-subroutine, PUSH, or a return-from-subroutine, POP, instruction from the CCU microprogram word.

Main memory poses an interesting set of problems to computer designers. Random access memory is typically slower than the computer control unit or arithmetic logic unit speed. And, on the other hand, read only memory may be much faster than the control process. The same set of problems is presented to the system by peripheral devices and processes. The computer control unit contends with these problems as well as the problems of synchronizing asynchronous events.

THE CONTROL SEQUENCE

The computer control unit contains an instruction register, microprogram storage and usually a microprogram register. Figure 2 presents a state diagram for a typical computer control unit. The first state of any processor must be an initialization sequence, regardless of its level of complexity or sophistication. The purpose of the initialization sequence is to place all of the system control storage elements in a known state such that control of the process can be started in an orderly manner. For example, registers, condition code, flag, and carry/link flip-flops are either preset to logic "1" or cleared to logic "0". Sometimes a sequence of events takes place such as the initialization of sets of register stacks or main memory. Also because some peripheral equipment may be involved that may be damaged by randomly changing states at its interface, very close attention must be given to the initialization process within the CCU state machine. A further requirement of this initialization process is that clock pulses must be withheld from the initialized hardware in some manner until the initialization procedure is completed.

The initialization sequence is usually started by one of three events: application of primary power to the system; either a programmed or operator generated "Master Reset" command; or an error that the state machine cannot recover from, but can detect. In a power-up generated initialization sequence, care must be given to the circuit that detects the event and generates the timed reset signal. The various power supply filters and loads must be considered as the state machine sequence should not be allowed to start until the entire power system is stable. Furthermore since some equipment and components may be damaged if they require multiple voltages that are not applied in the proper order, the computer control unit quite often is used to sequence the enabling of power supplies.

State "B" is the first computer minor cycle period. (A minor cycle is one primary clock period in length; characteristically, one microinstruction is executed. A major cycle is composed of one or more minor cycles and describes the completion of a macroinstruction or macroprocessors; i.e., "ADD" or "INTERRUPT"). During this state, the processor may be interrupted, halted, paused, or, in the absence of any of these requests, the computer control unit will fetch a macroinstruction from main memory and load it into the Instruction Register.

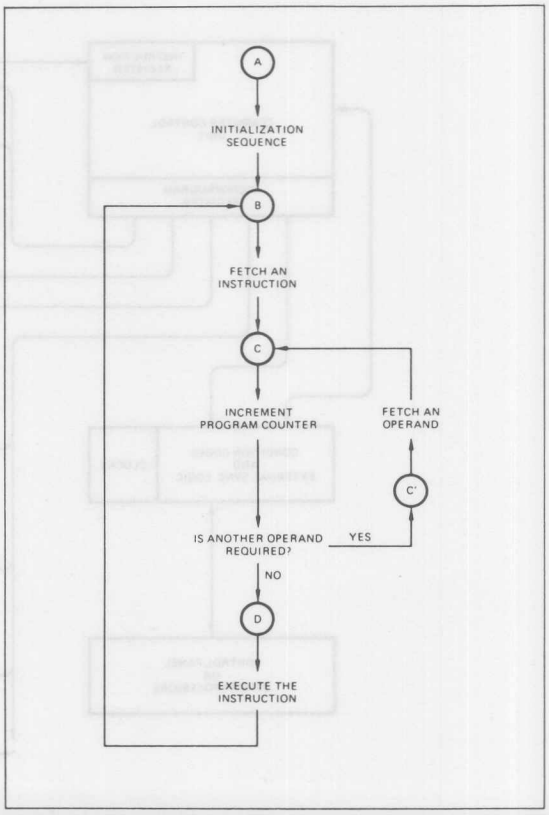


Figure 2. Simple Computer Control Unit State Transition Diagram.

Subsequently, during state "C" the Program Counter will be incremented and the instruction previously fetched will be decoded. If another operand is required for the current instruction, state "C"' will be executed the necessary number of times, and the operands will be loaded into the appropriate registers until the requirements of the instruction have been satisfied.

The last state, "D", is where the macroinstruction is executed. As in all of the other states in the process, the instruction execution state may require one or more microinstruction cycles. Having completed this state, control of the CCU will revert to state "B" microcode after a microinstruction branch to the beginning of that sequence has been effected.

CCU ARCHITECTURE

A functional representation of a computer control unit is presented in Figure 3. To aid in the diagram reference process, the major subsystem components are labeled with the designations C_1 to C_8.

The Instruction Register, C_1, receives the instruction from main memory via the data bus. The width of the register is generally the same as the memory word and data bus width to conserve processor overhead time. That is, if one clock period is necessary to fetch an instruction and one clock period is used to execute the instruction, that is a much more efficient use of computer time than requiring two or more clock periods to fetch the instruction and only one clock period to execute it. Any time required by the processor over and above the instruction execution time is considered overhead.

An instruction is broken down into two or more fields: the "Op Code," and one or more operands. An Op Code (Operation Code) is the instruction itself. The operands are data used by the computer control unit in the execution of the instruction. For example, an operand might be the number of a selected register, a variable to be compared to the accumulator, the address of an input/output port, etc.

Because the operand may be used as data, it must be presented to the data bus via an open collector or three-state transmitter. The operand and its subfields must also be distributed to the other computer subsystems that is serves, such as, the register selectors in the arithmetic/logic unit. The decoding and use of the Op Code, however, is not as obvious conceptually or from an implementation standpoint.

There is usually more than one microinstruction per macroinstruction. And, different classes of macroinstructions almost always require a different number of microprogram steps. The designer that is interested in a computer with only a few instructions may eliminate some hardware by using the Op Code from the Instruction Register, C_1, directly as the starting address of the Microprogram ROM, C_4. This is not only

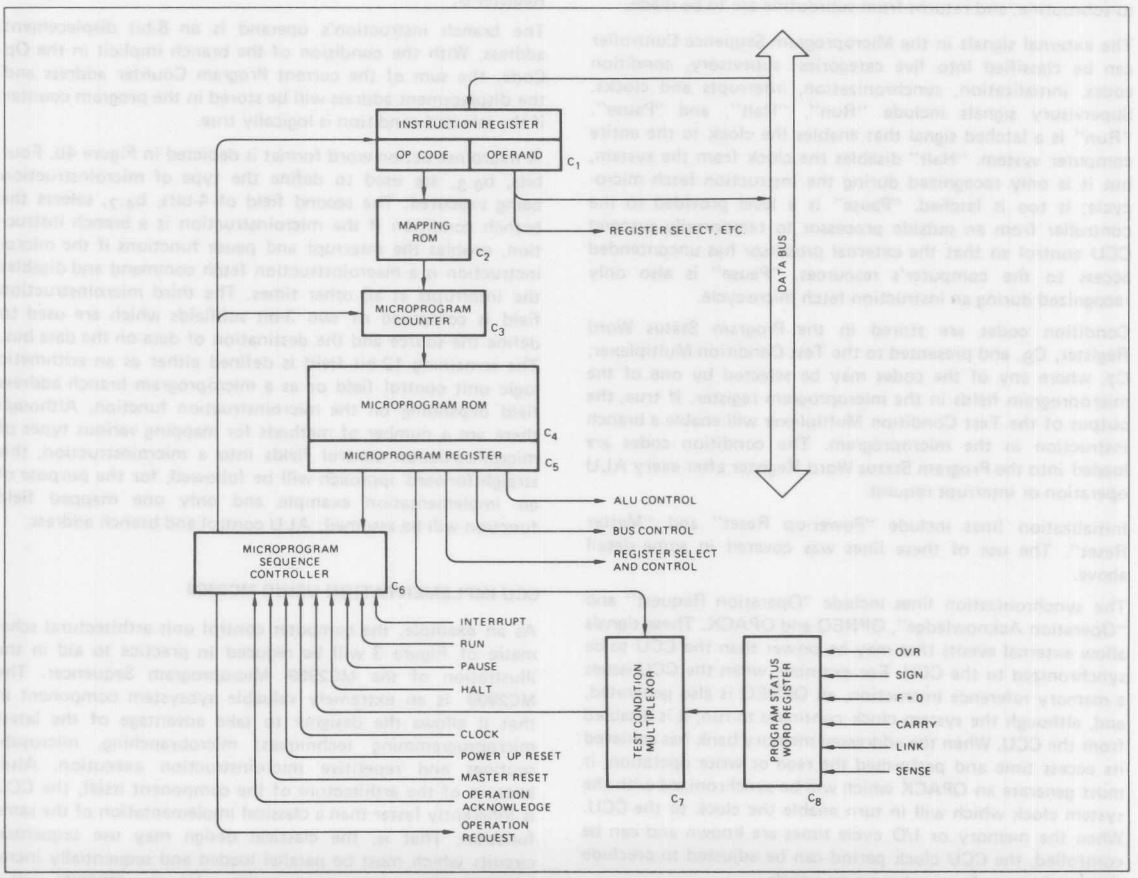


Figure 3. Computer Control Unit Architectural Schematic.

wasteful of Op Codes and inflexible, but, it means that any change in the instruction set or microprogram perturbs the entire system. To avoid this problem a Mapping ROM may be used.

The output of the Mapping ROM, C_2, should be wider than the Op Code field that is used as the address input. This allows a greater range of starting address for the Microprogram ROM, C_4. Because ROM/PROM field widths are typically 4 or 8-bits wide, a reasonable choice of width for the Mapping ROM with an 8-bit Op Code is 12-bits. The starting address is loaded into the Microprogram Counter, C_3, which points to the first microinstruction in the Microprogram ROM. When the output of the Microprogram ROM stabilizes, it is loaded into the Microprogram Register, C_5.

The use of the Microprogram Register in this manner is called pipelining. A pipeline register speeds up a state machine of this sort because it allows the address of the Microprogram ROM to be changed, and its output to settle, while the current microinstruction is being presented to the computer hardware from the Microprogram Register.

The Microprogram Sequence Controller, C_6, has two basic functions: it synchronizes events external to the CCU with the CCU, and it uses the output of the Test Condition Multiplexer to determine whether or not microprogram branches, jumps-to-subroutine, and returns-from-subroutine are to be made.

The external signals in the Microprogram Sequence Controller can be classified into five categories: supevisory, condition codes, initialization, synchronization, interrupts and clocks. Supervisory signals include "Run", "Halt", and "Pause". "Run" is a latched signal that enables the clock to the entire computer system. "Halt" disables the clock from the system, but it is only recognized during the instruction fetch microcycle; it too is latched. "Pause" is a level provided to the controller from an outside processor to temporarily suspend CCU control so that the external processor has uncontended access to the computer's resources. "Pause" is also only recognized during an instruction fetch microcycle.

Condition codes are stored in the Program Status Word Register, C_8, and presented to the Test Condition Multiplexer, C_7, where any of the codes may be selected by one of the microprogram fields in the microprogram register. If true, the output of the Test Condition Multiplexer will enable a branch instruction in the microprogram. The condition codes are loaded into the Program Status Word Register after every ALU operation or interrupt request.

Initialization lines include "Power-up Reset" and "Master Reset". The use of these lines was covered in some detail above.

The synchronization lines include "Operation Request" and "Operation Acknowledge", OPREQ and OPACK. These signals allow external events that may be slower than the CCU to be synchronized to the CCU. For example, when the CCU issues a memory reference instruction, an OPREQ is also generated, and, although the system clock continues to run, it is disabled from the CCU. When the addressed memory bank has achieved its access time and performed the read or write operation, it must generate an OPACK which will be synchronized with the system clock which will in turn enable the clock to the CCU. When the memory or I/O cycle times are known and can be controlled, the CCU clock period can be adjusted to preclude the requirement for synchronizing signals.

An interrupt may occur at any time, however, it is only recognized at an instruction fetch microcycle. At the time the interrupt is allowed, the priority encoded interrupt vector is jammed into the Program Status Word Register and the Microprogram ROM address is forced to the interrupt service routine address. When the interrupt has been serviced, the Microprogram Counter is returned to instruction fetch minor cycle address and processing resumes.

CCU INSTRUCTIONS

As implied earlier, there are two types of instructions recognized within the CCU, machine language or macroinstructions, and random logic replacement or microinstructions. Macroinstructions reside in main memory, are fetched and loaded into the instruction register and then decoded into microinstructions which directly control the computer's resources.

An example of two different types of macroinstructions may be seen in Figure 4a. A 16-bit instruction was defined with a constant length Op Code defined in the least significant 8 bits of the instruction. The remainder of the instruction word, bits 8 through 15, will be defined as a function of instruction type.

The register-to-register instruction has two operand fields that select the source and destination register, Register A and Register B, respectively. That is, the result of an arithmetic/logic function with Registers A and B will be stored in Register B.

The branch instruction's operand is an 8-bit displacement address. With the condition of the branch implicit in the Op Code, the sum of the current Program Counter address and the displacement address will be stored in the program counter if the selected condition is logically true.

A microinstruction word format is depicted in Figure 4b. Four bits, b_{0-3}, are used to define the type of microinstruction being executed. The second field of 4-bits, b_{4-7}, selects the branch condition if the microinstruction is a branch instruction, enables the interrupt and pause functions if the microinctruction is a macroinstruction fetch command and disables the interrupts at all other times. The third microinstruction field is composed of two 3-bit subfields which are used to define the source and the destination of data on the data bus. The remaining 12-bit field is defined either as an arithmetic logic unit control field or as a microprogram branch address field depending on the microinstruction function. Although there are a number of methods for mapping various types of microinstruction control fields into a microinstruction, this straightforward approach will be followed, for the purpose of an implementation example and only one mapped field function will be assumed: ALU control and branch address.

CCU IMPLEMENTATION USING MC2909

As an example, the computer control unit arthitectural schematic of Figure 3 will be reduced in practice to aid in the illustration of the MC2909 Microprogram Sequencer. The MC2909 is an extremely valuable sybsystem component in that it allows the designer to take advantage of the latest microprogramming techniques; microbranching, microsubroutines and repetitive microinstruction execution. Also, because of the arthitecture of the component itself, the CCU is inherently faster than a classical implementation of the same function. That is, the classical design may use sequential circuits which must be parallel loaded and sequentially incremented with separate clock pulses, while the MC2909 uses a combinational incrementor outside of the microprogram address bus which is transferred to the microprogram counter

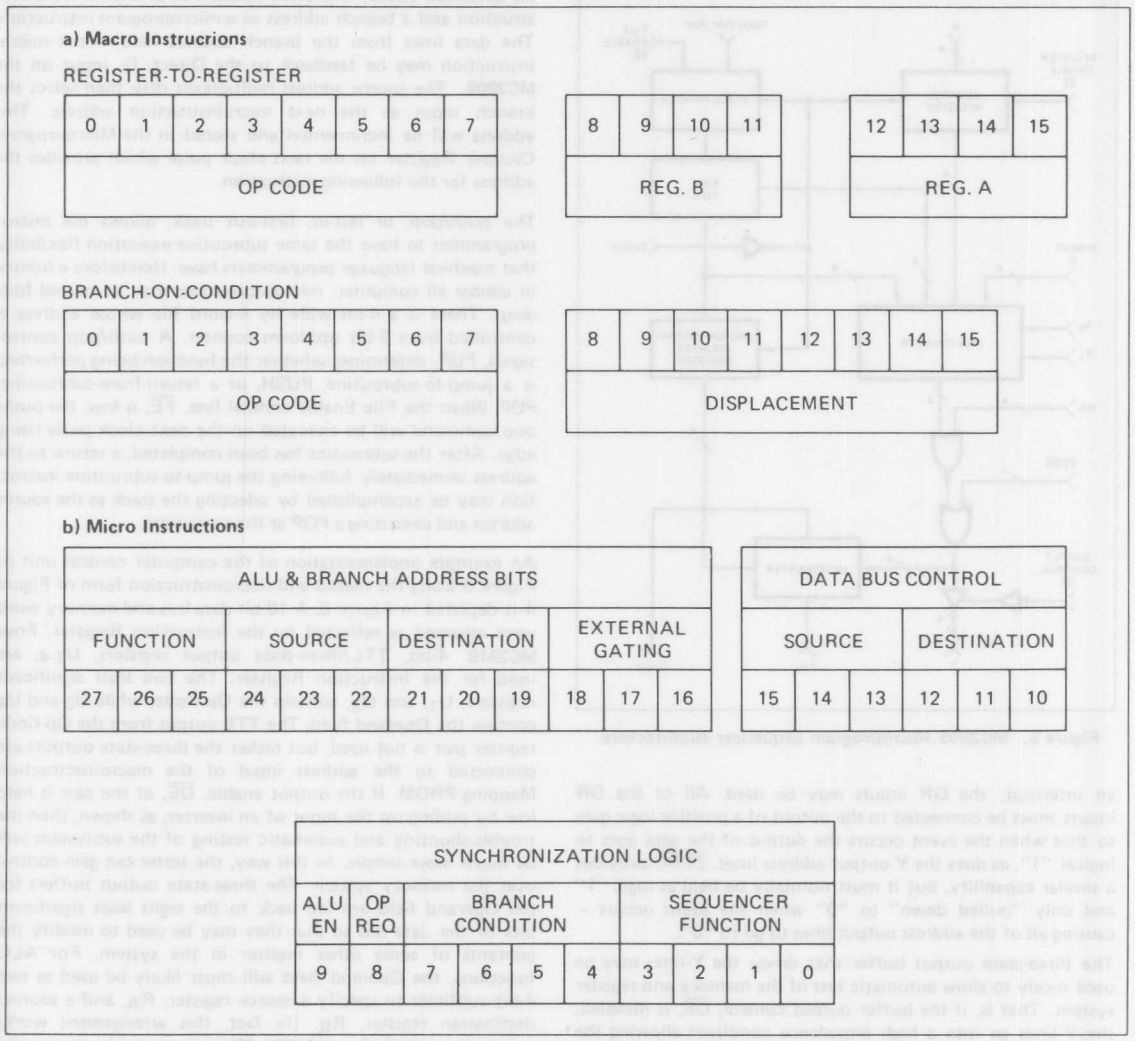


Figure 4. Example Macro and Micro-Instruction Fields.

on the rising edge of the clock pulse. A detailed specification of the MC2909 is provided in the expanded data sheet and its internal architectural rendering is reproduced here in Figure 5.

The purpose of the MC2909 is to present an address to the microprogram ROM such that a microinstruction may be fetched and executed. In referencing Figure 5, there are four sources of address information available: and Address Register, a Microprogram Counter Register, a Direct or branch input, and a subroutine stack. The address source is chosen by using the one-of-four address multiplexer select lines, S_0 and S_1. The selected address may then be modified by the OR input lines or the $\overline{ZERO}$ input function before it is presented at the Y address output lines through a three-state buffer.

The OR input lines may be used in one of two manners. Selected OR inputs may be placed at logic "1" which will provide the logical OR of the selected address source and the OR input lines at the Y output. This allows the address to be "masked". If a microprogram instruction of the Skip or Branch classes is being executed and the microinstruction is aligned on an even address microprogram ROM word (the least significant address bit is "0"), then the least significant OR input may be controlled by an external test condition multiplexer. If the result of the conditional test was logically false, then the least significant bit may be modified to avoid the execution of the Branch or Skip instruction. All of the unused inputs must be tied to ground. Similarly, if the 2, 3, 4, or n-least significant bits of the selected address are "0", the associated OR input lines may be modified for an extended address range skip capability.

Sometimes, in a state machine like a computer control unit, it is desirable to easily get to a predefined state, or address. For instance, if the machine has just been turned on and it is necessary to perform an initialization sequence or a real-time event occurs where the processor control is required but the on-going process information may not be destroyed, such as

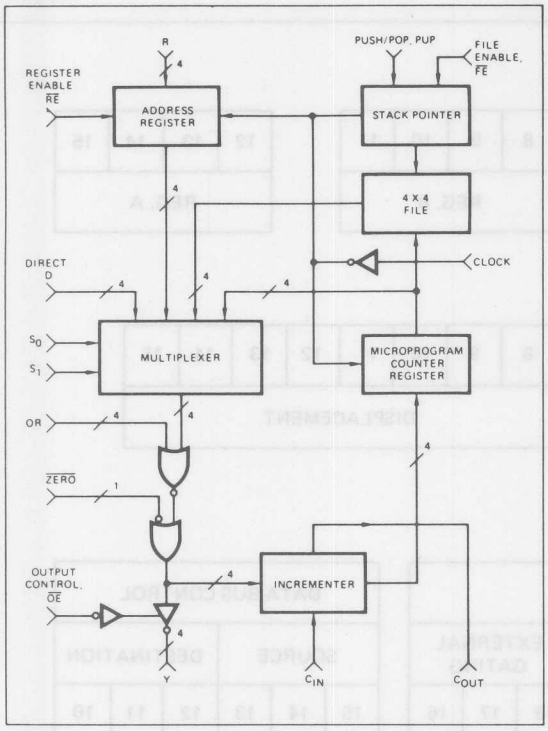


Figure 5. MC2909 Microprogram Sequencer Architecture.

an interrupt, the OR inputs may be used. All of the OR inputs must be connected to the output of a positive logic gate so that when the event occurs the output of the gate goes to logical "1", as does the Y output address lines. $\overline{ZERO}$ provides a similar capability, but it must normally be held at logic "1" and only "pulled down" to "0" when the event occurs – causing all of the address output lines to go to "0".

The three-state output buffer that drives the Y-lines may be used nicely to allow automatic test of the memory and register system. That is, if the buffer output control, $\overline{OE}$, is disabled, the Y-lines go into a high impedance condition allowing the automatic tester's output lines to be connected directly across the outputs. This capability also allows multiple processors to share the same memory by enabling only one processor's Y-bus at a time.

The Address Register, as well as all other storage devices on the MC2909, is parallel loaded from the R inputs when the register enable line, $\overline{RE}$, is low on a positive going clock transition. This is a good register to use when entering the starting address of a microprogram. If selected, the contents of the register are not only presented to the Y outputs, but also to the Incrementer.

The Incrementer is a full-adder provided with an off-chip carry-in signal, C_{IN}, and a off-chip carry-out signal, C_{OUT}, allowing multiple MC2909's to be cascaded. The output from the Incrementer is connected to a parallel load input on the Microprogram Counter Register where it is loaded on the rising edge of the next clock pulse. If the Microprogram Counter is selected as the source address by subsequent microinstructions, it will be incremented by each succeeding clock pulse, thereby stepping through the microprogram.

As described above, it is often valuable to provide a branch instruction and a branch address in a microprogram instruction. The data lines from the branch address field in the microinstruction may be feedback to the Direct, D, input on the MC2909. The source address multiplexer may then select the branch input as the next microinstruction address. This address will be incremented and stored in the Microprogram Counter Register on the next clock pulse which provides the address for the following instruction.

The push/pop, or last-in, first-out stack, allows the microprogrammer to have the same subroutine execution flexibility that machine language programmers have. Heretofore a luxury in almost all computer, microsubroutines may be nested four deep. There is a 4-bit wide by 4-word file whose address is controlled by a 2-bit up/down counter. A push/pop control signal, PUP, determines whether the function being performed is a jump-to-subroutine, PUSH, or a return-from-subroutine, POP. When the File Enable control line, $\overline{FE}$, is low, the push/pop command will be executed on the next clock pulse rising edge. After the subroutine has been completed, a return to the address immediately following the jump-to-subroutine instruction may be accomplished by selecting the stack as the source address and executing a POP at the same time.

An example implementation of the computer control unit of Figure 3 using the macro and microinstruction form of Figure 4 is depicted in Figure 6. A 16-bit data bus and memory word were assumed as reflected by the Instruction Register. Four MC2918 4-bit, TTL/three-state output registers, $U_{1\text{-}4}$, are used for the Instruction Register. The two least significant registers, U_1, and U_2, contain the Op Code, while U_3 and U_4 contain the Operand field. The TTL output from the Op Code register pair is not used, but rather the three-state outputs are connected to the address input of the macroinsctruction Mapping PROM. If the output enable, $\overline{OE}$, of the pair is held low by pulling up the input of an inverter, as shown, then the trouble-shooting and automatic testing of the subsystem will be much more simple. In this way, the tester can gain control over the memory system. The three-state output buffers for the Operand field are fed back to the eight least significant bits of the data bus so that they may be used to modify the contents of some other register in the system. For ALU functions, the Operand field will most likely be used as two 4-bit subfields to specify a source register, R_A, and a source/destination register, R_B. (In fact, this arrangement works extremely well if the MC2901 Microprocessor is employed.) The TTL outputs are used for R_A and R_B data.

The mapping PROM's that were used, selected for their speed and architecture, are three Am29761's in parallel. With a memory configuration of 256 words by 4 bits, each of the 256 potential Op Codes has a unique 12-bit starting address which provides the designer with a lot of flexibility for his initial design and an unusually easy task of adding more instructions at a later date.

In turn, the Mapping PROM outputs are connected to the address register inputs, $R_{0\text{-}11}$, of the three MC2909 Microprogram Sequencers. The Microprogram Sequencer outputs, $Y_{0\text{-}11}$, provide the address inputs for seven Am29761 Microprogram PROM's. (The output enable lines of the MC2909's should be controlled in the same manner as the Instruction Register outputs.) Although only 256 words of microprogram storage are shown, up to 4096 words may be implemented if necessary. Furthermore, if more than 28-bit microinstruction words are required for the user's task they may be added as necessary.

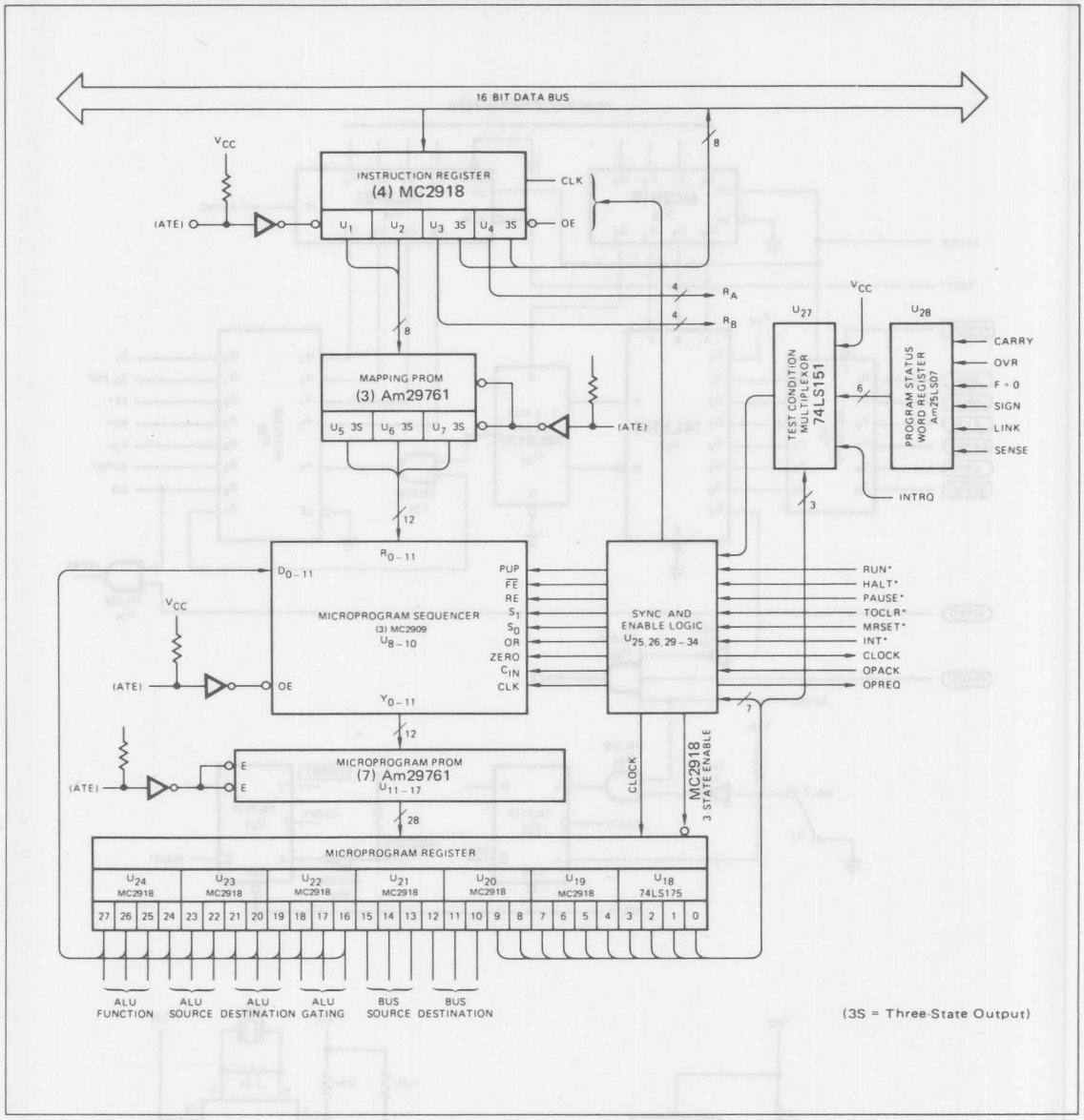


Figure 6.

The Microprogram Register consists of one 74LS175 register, U_{18}, and six MC2918 registers, U_{19-24}. The ten least significant bits are used by the synchronization and enable logic. The most significant 12 bits are used for either Microsequencer branch address (the TTL outputs of the MC2918's) or for control of the ALU (the 3-state lines are used). As shown, the ALU control bit fields are specified to control four MC2901's and perform all of the necessary external gating and bit manipulation. The remaining six central bits are provided for data bus source and destination controls and the 3-state outputs are used. Whenever the processor is running the 3-state output enable lines are held low, enabling the output. If the processor has been Paused, austensibly for direct memory access, the outputs are disabled so that an external or peripheral processor can gain access to the control line.

The Sync. and Enable Logic is relatively complex and may be shown better in Figure 7. The two least significant registers in the Microprogram registers, U_{18} and U_{19}, are at the top of the page. In addition, the remaining 2 bits from the Microprogram Register that are used here, ALUEN* and OPREQ, are shown with rectangular boxes around them so that they are easy to see. The control and status bits that emanate from portions of the computer other than the CCU are shown enclosed in ovals. All other signals are generated or used with the CCU.

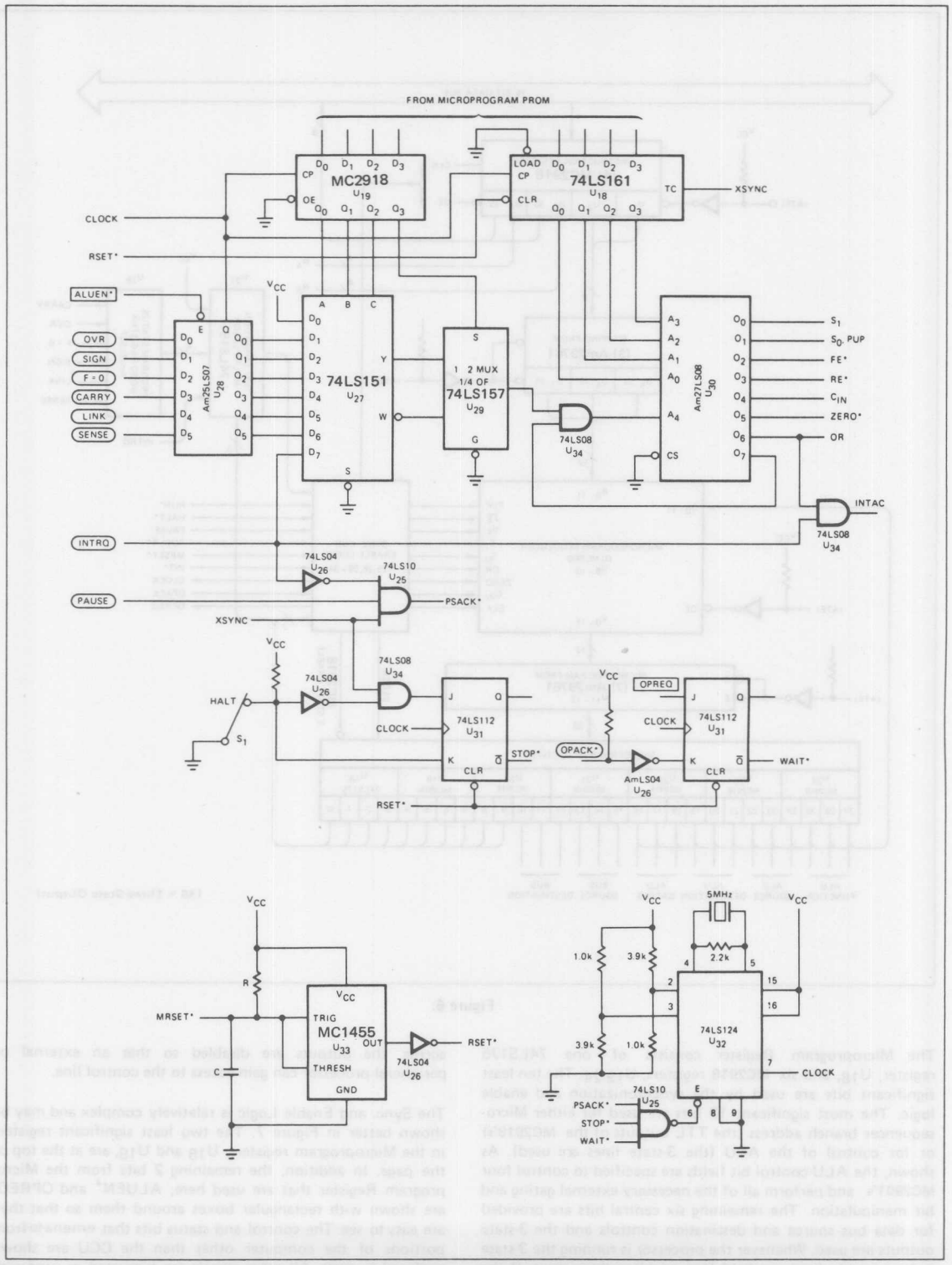


Figure 7. Sync. and Enable Logic Diagram.

The 4 bits stored in Microprogram Register U_{18} provide the Microprogram Sequencer Function instruction. A **74LS161** was selected for this register because it was synchronous, has an asynchronous clear (enabling power-up reset) and is low-power Schottky. These 4 bits provide the least significant address bits of an Am27S08 32-word by 8-bit PROM, as well as providing an external event synchronizing signal enable, XSYNC. With XSYNC at logic "1", the external processes that use PAUSE will be enabled allowing direct memory access.

The Am27S08 PROM has eight open collector outputs (that must have pull-up resistors added). Seven of the output signals go directly to the 3 sets of **MC2909** control lines. The eighth output line is fed back to enable a gate that drives the fifth PROM address line. The other signal at the AND-gate, U_{34}, is the test condition enable line.

The test condition signal is the result of selecting one of eight processor condition signals using the **74LS151** multiplexer, U_{27}. U_{27} has two outputs, the selected signal and its complement. One of these two signals is selected using (1/2) 74LS51, U_{29}, as the multiplexer. The 4-bit select signal is stored in Microprogram Register U_{19}. Notice that one of the condition code multiplexer's inputs is tied to V_{CC} which provides for an unconditional branch if the entire register, U_{19}, is "0". Six of the condition codes (from the ALU) are stored in a **74LS157** register. Everytime an ALU function is selected and clocked, as denoted by ALUEN*, the current value of the condition codes are clocked into U_{28}. The eighth condition code bit is the interrupt request signal INTRQ which is latched externally.

Before proceeding with the rest of the synchronization and enable circuitry, let us consider the programmability aspects of this portion of the state machine. A table of desirable Microprogram Sequencer functions is provided in Figure 8. In fact, this table is also the memory map for the Microprogram Sequencer PROM, U_{30}. Entries are made in the table by U_{30} address value. The first 16 entries have the test condition address bit equal to "0". These are the primary instructions; they enable the **MC2909** functions. Of the primary instructions there are only four that have O_7, the test condition enable bit set. Three of these are branch instructions and for these microinstructions any condition code may be specified in register U_{20} except interrupt request, INTRQ. The remaining instruction relates to the macroinstruction fetch process (Figure 2) and only during this microcycle may the CCU be interrupted or paused, as data or instructions moved to system registers under microprogram control may be lost if the microprocess is disturbed. Processed interrupts, by definition, disturb the microprocess and may usurp control of any of the computer's resources.

The only instance when there will be a secondary instruction defined, is when bit O_7, equals "1" such that potentially A_5 equals "1". In the event that the condition code test was successful, the four secondary instructions defined in Figure 8 will be executed.

If an interrupt was generated during an instruction fetch, the OR output, U_{30}, O_6, will assume logic "1". This signal will be logically ANDed by U_{34} to generate the interrupt acknowledge signal, INTAC. INTRQ, or the absence of the granting of the external synchronization signal, XSYNC, that is generated during the instruction fetch, will preclude the pause acknowledge signal, PSACK*.

An attempt to Halt the processor using an external switch S_1 will also be denied unless the current microinstruction cycle is a macroinstruction fetch. Starting the processor by moving switch S_1 to the RUN position will always be granted and synchronized by U_{31} because by definition the processor stopped previously at an instruction fetch cycle which is also the first state which must be executed when the processor is turned on. If STOP, the Q output of U_{31} is logic "0", the processor will HALT.

Provision has been made to synchronize the relatively fast CCU with relatively slow memory or input/output functions. If a microinstruction causes memory or I/O reference, the microprogrammer must set the OPREQ bit true. This signal is latched-up in the second half of flip-flop U_{31} and stops the processor (WAIT = "0") until the address device acknowledges the operation is complete or the data is ready by pulling down the OPACK* line. U_{31} synchronizes the event and restarts the processor.

We have discussed a "hard" processor interrupt event, INTRQ, a "soft" interrupt (one where the state of the processor is not disturbed, but operation is suspended), PAUSE, starting and stopping the processor, RUN and HALT, and synchronizing the processor with external events, OPREQ/OPACK*. Let us consider the system clock and system initialization.

A good choice for a system oscillator is the 74LS124, U_{32}. This device is a dual voltage controlled oscillator whose timing may be derived either by a series mode, fundamental frequency crystal, or an RC timing circuit. For high speed digital circuits, it is necessary to use a crystal. The output of the oscillator is free running and is presented to an on-chip pulse synchronizer controlled by the enable line on the chip. No partial pulses can be passed by the synchronizer so that control of the oscillator's enable line may be asynchronous. A PAUSE, HALT, or OPREQ will cause the gated oscillator to shut off until the process is allowed to restart.

Initialization of a state machine is very important. When the power is applied to the system, the system must be disabled until all of the power supply filters have charged and the regulators stabilized. Also, critical storage devices must be preset to a known state. To accomplish this, a **MC1455** timer circuit, U_{33} with output buffer, U_{26}, is used. The RC value should yield a time-out circuit increment greater than 100msec in most systems (t = 1.1RC for this device). Also, by using an external switch or open collector gate to ground a MRSET*, the entire system may be master reset without cycling the power supply off and then on again. Other than clearing flip-flops, the output of the initialization circuit RSET*, clears the Microprogram Register U_{19}. By referring to Figure 8 again, it can be seen that Microsequencer Function $A_{0\text{-}3}$ = "0" provides system initiation in that U_{30} output bit O_5, ZERO*, is low, thereby setting the initial microprogram address to zero and incrementing from there. This allows an initialization microprogram to be stored in the bottom of memory.

The ability to execute the same microinstruction a number of times was alluded to previously. Generally the value of this capability lies outside of the CCU. As an example, let us reconsider the macroinstruction format for a Register-to-Register instruction (Figure 4). If the Op Code is a Shift or Rotate instruction, it would be desirable to allow the programmer to move the data word over a range of 1 to 16-bit positions with a single instruction rather than having to execute the same instruction many times. Since there are two Operand subfields, R_A and R_B, let us define the 4-bit value in R_A as the number of bit positions we wish to move the

	MICROPROGRAM SEQUENCER FUNCTION					FUNCTION DESCRIPTION	MICROPROGRAM SEQUENCER CONTROL							
							O_7	O_6	O_5	O_4	O_3	O_2	O_1	O_0
	A_4	A_3	A_2	A_1	A_0		TEST ENABLE	OR	$\overline{ZERO}$	C_{IN}	$\overline{RE}$	$\overline{FE}$	S_0, PUP	S_1
TEST CONDITION DISABLED OR FALSE	0	0	0	0	0	Initialize System	L	X	L	H	H	H	X	X
	0	0	0	0	1	Branch Test	H	L	H	H	H	H	L	L
	0	0	0	1	0	Jump to Subroutine Test	H	L	H	H	H	H	L	L
	0	0	0	1	1	Return from Subroutine	H	L	H	H	H	H	L	L
	0	0	1	0	0	Execute Program	L	L	H	H	H	H	L	L
	0	0	1	0	1	External Carry Control	L	L	H	*	H	H	L	L
	0	0	1	1	0	This Group Undefined	L	L	H	H	H	H	L	L
	0	0	1	1	1		L	L	H	H	H	H	L	L
	0	1	0	0	0	This Group Undefined	L	L	H	H	H	H	L	L
	0	1	0	0	1		L	L	H	H	H	H	L	L
	0	1	0	1	0		L	L	H	H	H	H	L	L
	0	1	0	1	1		L	L	H	H	H	H	L	L
	0	1	1	0	0	This Group Undefined	L	L	H	H	H	H	L	L
	0	1	1	0	1		L	L	H	H	H	H	L	L
	0	1	1	1	0	Load Mapped (Starting) Address	L	L	H	H	L	H	H	L
	0	1	1	1	1	Fetch Instruction	H	L	H	H	H	H	L	L
TEST CONDITION ENABLED AND TRUE	1	0	0	0	0	This State Undefined	L	–	–	–	–	–	–	–
	1	0	0	0	1	Execute Branch	H	L	H	H	H	H	H	H
	1	0	0	1	0	Execute Jump	H	L	H	H	H	L	H	H
	1	0	0	1	1	Execute Return	H	L	H	H	H	L	L	H
	1	0	1	0	0	This State Undefined	L	–	–	–	–	–	–	–
	1	0	1	0	1		L	–	–	–	–	–	–	–
	1	0	1	1	0		L	–	–	–	–	–	–	–
	1	0	1	1	1		L	–	–	–	–	–	–	–
	1	1	0	0	0	This State Undefined	L	–	–	–	–	–	–	–
	1	1	0	0	1		L	–	–	–	–	–	–	–
	1	1	0	1	0		L	–	–	–	–	–	–	–
	1	1	0	1	1		L	–	–	–	–	–	–	–
	1	1	1	0	0	This State Undefined	L	–	–	–	–	–	–	–
	1	1	1	0	1		L	–	–	–	–	–	–	–
	1	1	1	1	0		L	–	–	–	–	–	–	–
	1	1	1	1	1	Service Interrupt or Pause	H	H	H	H	H	H	X	X

*Value of this Bit depends on Logic Implementation. See Text.

Figure 8. Microsequencer Function Table.

data, and R_B as the general purpose register that will be effected. (The additional hardware to implement the circuit is shown in Figure 9.)

The value in R_A must be parallel loaded into a 4-bit binary counter that has a terminal count flag, TC (when $Q_0 = Q_1 = Q_2 = Q_3 = 1$, TC = 1), such as the 74LS163. The MC2909 control signal RE^*, that loads the Address Register, must also be applied to the 74LS163 signal $LOAD^*$. CLOCK is merely the system clock, and MLTEN is a signal that must be supplied by the microprogram to enable this function. MLTEN and TC are connected to an open-collector AND-gate which pulls down the MC2909 carry-in line until terminal count has been achieved. As a result, the microprogram address does not change until TC equals "1" and then C_{IN} equals "1" which increments the microprogram counter causing the next instruction to be executed. The number of reasons for using this feature are almost unlimited, as are the means to implement the function.

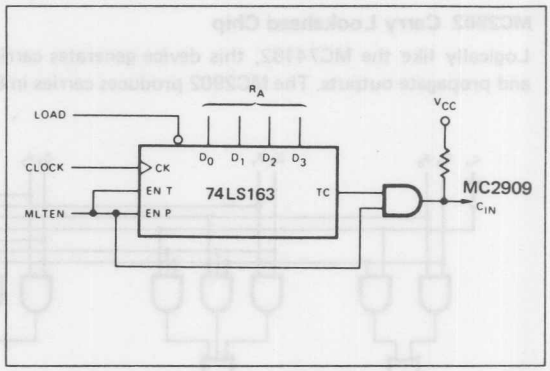
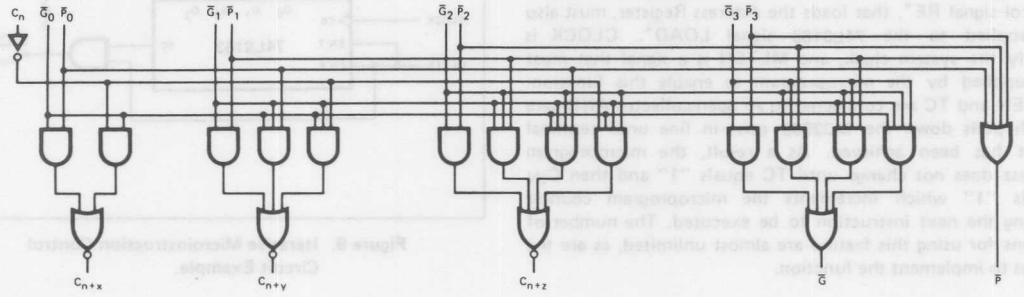


Figure 9. Iterative Microinstruction Control Circuit Example.

PRODUCT PREVIEW

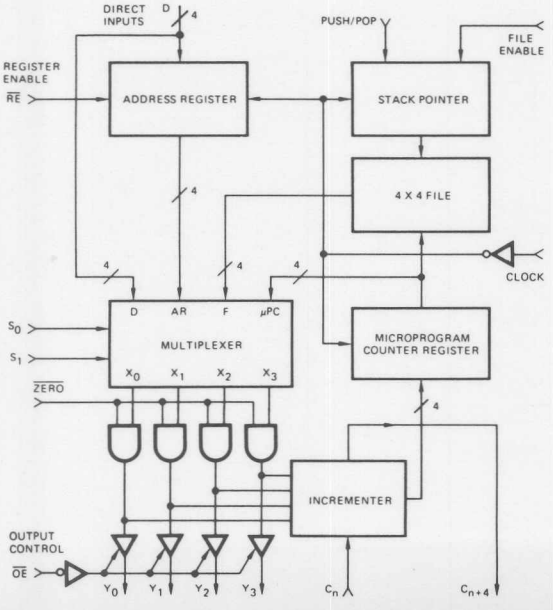
MC2902 Carry Lookahead Chip

Logically like the MC74182, this device generates carries for three MC2901's from the generate and propagate outputs. The MC2902 produces carries in 8ns from the G and P inputs.



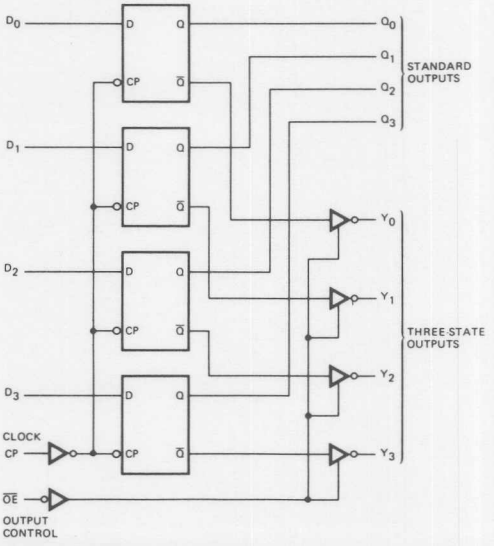
MC2911 Minimicroprogram Sequencer

The MC2911 contains nearly all the power of the MC2909 in a space and cost saving 20-pin package. Delay from clock to output is typically 40ns.



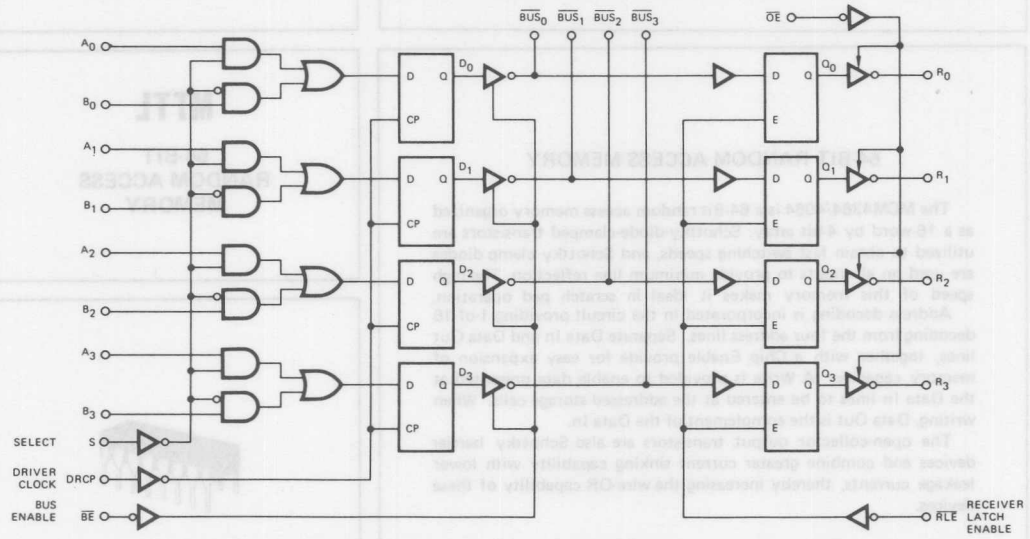
MC2918 One-by-Two Port Register

A four-bit register, one set of inputs; two sets of outputs – three-state and regular Schottky TTL. Ideal for status registers, data bus interfaces.



MC2905 and MC2915 LSI Bus Transceivers

The MC2905 is a four-bit bus transceiver designed to drive a 100mA load on an open collector bus. The MC2915 is an identical circuit, designed to drive a three-state bus. They include a two-port input register and a latch on the receiver outputs. The latch is three-state. Delay from clock to bus is typically 21ns; from bus to receiver output is typically 18 ns.

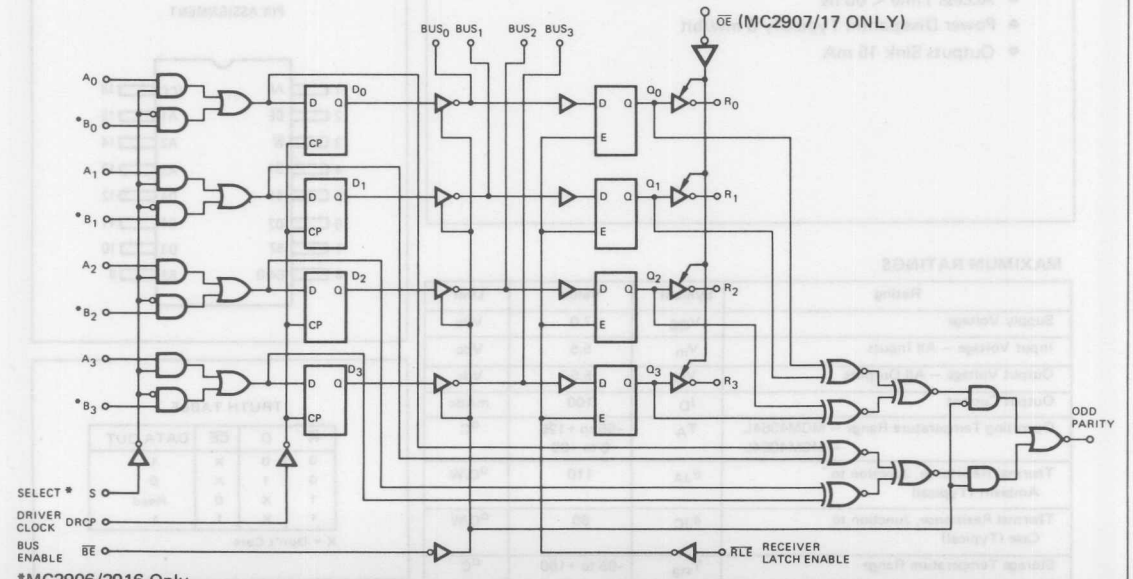


MC2906 and MC2916 LSI Bus Transceivers

Like the MC2905 and MC2915, but includes a parity generator/checker on chip.

MC2907 and MC2917 LSI Bus Transceivers

Like the MC2906 and MC2916, but has only one data input port on the driver register. Fits in space-saving, 20-pin package.



*MC2906/2916 Only.

MOTOROLA Semiconductors

BOX 20912 • PHOENIX, ARIZONA 85036

MCM4364L
MCM4064L

MTTL

64-BIT RANDOM ACCESS MEMORY

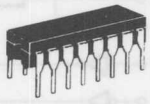
64-BIT RANDOM ACCESS MEMORY

The MCM4364/4064 is a 64-Bit random access memory organized as a 16-word by 4-bit array. Schottky-diode-clamped transistors are utilized to obtain fast switching speeds, and Schottky clamp diodes are used on all inputs to provide minimum line reflection. The high speed of this memory makes it ideal in scratch pad operation.

Address decoding is incorporated in the circuit providing 1-of-16 decoding from the four address lines. Separate Data In and Data Out lines, together with a Chip Enable provide for easy expansion of memory capacity. A Write is provided to enable data presented at the Data In lines to be entered at the addressed storage cells. When writing, Data Out is the complement of the Data In.

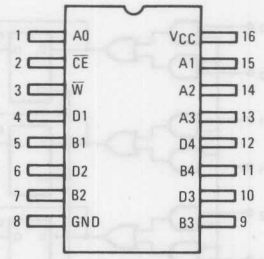
The open-collector output transistors are also Schottky barrier devices and combine greater current sinking capability with lower leakage currents, thereby increasing the wire-OR capability of these devices.

- Both Minimum and Maximum Access Times Specified
- Binary Addressing
- Chip Enable for Memory Expansion
- Outputs May Be "Wire ORed"
- Logic Levels Compatible with MDTL and All MTTL Families
- Low-Voltage Input Clamp Diodes
- Access Time < 60 ns
- Power Dissipation Typically 6 mW/bit
- Outputs Sink 15 mA



CERAMIC PACKAGE
CASE 620

PIN ASSIGNMENT



TRUTH TABLE

W	D	CE	DATA OUT
0	0	X	1
0	1	X	0
1	X	0	Read
1	X	1	1

X = Don't Care

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	7.0	Vdc
Input Voltage – All Inputs	V_{in}	5.5	Vdc
Output Voltage – All Outputs	V_D	5.5	Vdc
Output Current	I_D	100	mAdc
Operating Temperature Range – MCM4364L – MCM4064L	T_A	-55 to +125 0 to +85	°C
Thermal Resistance, Junction to Ambient (Typical)	θ_{JA}	110	°C/W
Thermal Resistance, Junction to Case (Typical)	θ_{JC}	60	°C/W
Storage Temperature Range	T_{stg}	-65 to +160	°C

DC ELECTRICAL CHARACTERISTICS (T_A = -55 to +125°C for MCM4364L, 0 to +85°C for MCM4064L unless otherwise noted.)

Characteristic		Symbol	Min	Typ	Max	Unit
Input Forward Current — All Inputs (V_{IL} = 0.4 Vdc, V_{CC} = 5.25 Vdc)		I_{IL}	–	–	1.6	mAdc
Input Leakage Current — All Inputs (V_{IH} = V_{CC} = 5.25 Vdc)		I_{IH}	–	–	80	μAdc
Input Clamp Voltage — All Inputs (I_{IC} = -5.0 mAdc, V_{CC} = 4.75 Vdc, T_A = 25°C)		V_{IC}	–	–	-1.0	Vdc
Input Logic Levels — All Inputs (V_{CC} = 5.0 Vdc)	"0" Level	V_{IL}	–	–	0.8	Vdc
	"1" Level	V_{IH}	2.0	–	–	
Logic "0" Output Voltage — All Outputs (V_{CC} = 4.75 Vdc, I_{OL} = 15 mAdc)	MCM4364L	V_{OL}	–	–	0.50	Vdc
	MCM4064L		–	–	0.45	
Output Leakage Current — All Outputs (V_{CC} = V_{CEX} = 5.25 Vdc)		I_{CEX}	–	–	100	μAdc
Power Supply Current (V_{CC} = 5.25 Vdc, all inputs except $\overline{W}$ grounded)*		I_{CC}	–	–	105	mAdc
Input Capacitance — All Inputs (V_{in} = 2.0 Vdc, V_{CC} = 5.0 Vdc)		C_{in}	–	6.0	–	pF
Output Capacitance — All Outputs (V_{out} = 2.0 Vdc, V_{CC} = 5.0 Vdc)		C_{out}	–	8.0	–	pF

AC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc, T_A = 25°C)

Characteristic	Symbol	Min	Typ	Max	Unit
Access Times — Address Input or Chip Enable to Output Delay. (See Figures 1 and 2)	t_{A+B+}	15	–	60	ns
	t_{A-B-}	15	–	60	
	t_{A+B-}	15	–	60	
	t_{A-B+}	15	–	60	
	t_{CE+B+}	15	–	60	
	t_{CE-B-}	15	–	60	
Write Pulse Width (See Figures 1 and 3)	t_w	40	–	–	ns
Write Recovery Time (See Figures 1 and 3)	t_{wr}	–	–	50	ns
Address and Data Hold Time After Write (See Figures 1 and 3)	$t_{hold(A \pm W+)}$	5.0	–	–	ns

*Worst-case dc input condition.

BLOCK DIAGRAM

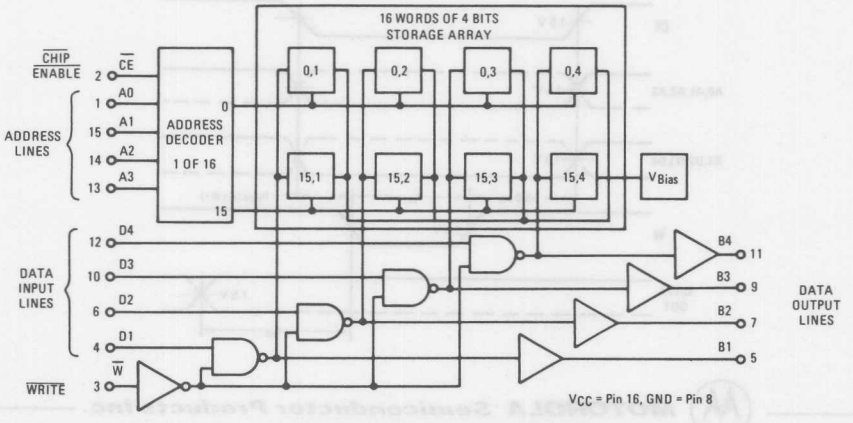


FIGURE 1 – SWITCHING TIME TEST
CONDITIONS AND LOAD CIRCUIT
V_{CC} = 5.0 Vdc, T_A = 25°C

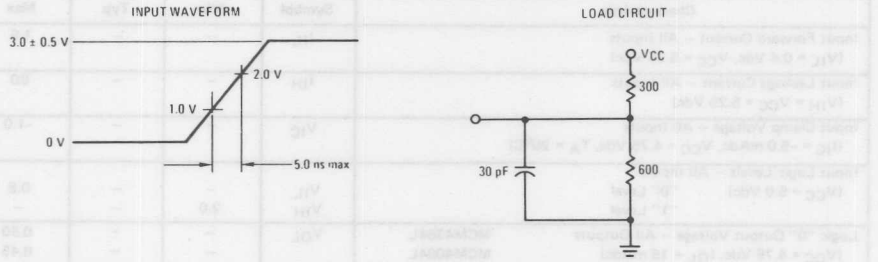


FIGURE 2 – ACCESS TIME DEFINITIONS

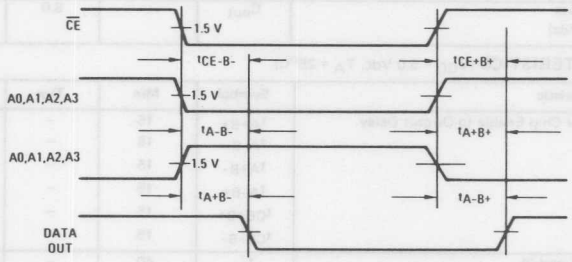
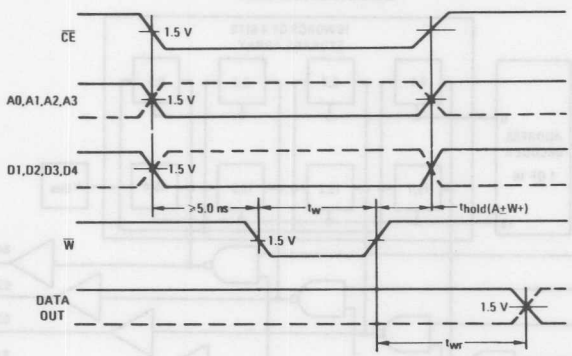


FIGURE 3 – WRITE TIME DEFINITIONS



TYPICAL ACCESS TIMES

ADDRESS TO OUTPUT

FIGURE 4 – VARIATIONS WITH SUPPLY VOLTAGE

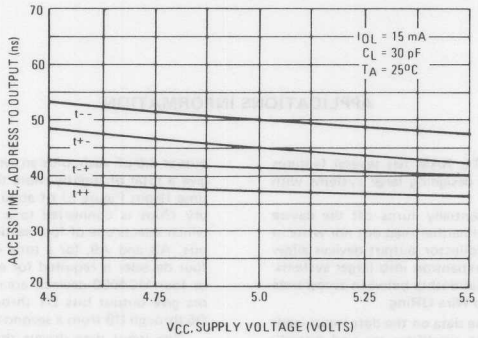
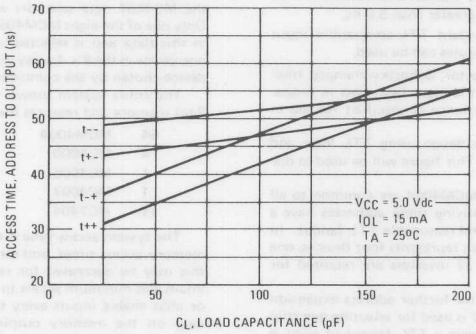
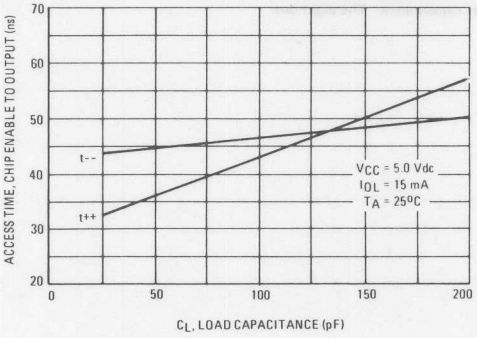


FIGURE 5 – VARIATIONS WITH LOAD CAPACITANCE



CHIP ENABLE TO OUTPUT

FIGURE 6 – VARIATIONS WITH LOAD CAPACITANCE



APPLICATIONS INFORMATION

The MCM4364/4064 64-bit MTTL RAM has several features which should be considered when designing large systems with this device.

1. $\overline{\text{Chip Enable}}$ This input essentially turns off the device when in the high state. The chip will neither read out nor write in data when disenabled. The open collector output devices allow Wire ORing or phantom logic for expansion into larger systems. Graphs in this data sheet show the relationship between access time and capacitance on the output due to Wire ORing.

2. Write The $\overline{W}$ input gates the data on the data input leads into the memory cells. The output amplifiers are tied directly to the write amplifiers and follow as the inverse of the input for either state of $\overline{CE}$ as long as $\overline{W}$ is low. There is, however, a "glitch" on the output lines just after $\overline{W}$ returns high or goes low. For proper writing into the memory, the address and data hold time after writing, $t_{hold(A\pm W+)}$, must be greater than 5.0 ns.

3. The address inputs have standard TTL-compatible input thresholds and standard TTL loading rules can be used.

The data outputs have open-collector, Schottky-clamped transistors and can be wire ORed with the incumbent cost in propagation delay. Typical output capacitance is specified for aid in system design.

Figure 7 shows a typical system design using TTL logic and the MCM4064 as a main frame store. This figure will be used to discuss several design considerations.

The four address inputs of the MCM4064 are common to all devices in the system. The gates driving these addresses have a fanout of eight, which is a typical and reasonable TTL fanout. In the diagram, each address driver shown represents four devices, one for each address. Thus, 8 x 4 or 32 inverters are required for address driving.

The chip enable inputs are used for further address expansion and a one-of-eight decoder (MC4006) is used for selecting one chip of eight with Wired OR outputs. Again a TTL fanout of eight is used. This chip enable decodings adds three address bits, A5, A6 and A7.

The memory chips are Wire-ORed in sections of eight. This figure of eight is used as a trade-off in decoding array time versus increase in access time due to output capacitance. The eight outputs at 8.0 pF each, plus an average of 5.0 pF for board capacitance give a total of slightly more than 100 pF for a total typical access time (from Figure 5) of about 50 ns. Each section of eight memory chips is connected to a one-of-four data selector (MC4000) which selects one of four sections. This adds a further two address bits, A8 and A9, for a total of 512 address locations. A one-of-four decoder is required for each data output and eight selectors, or four MC4000 devices, are required for the system. Four selectors give output bits D1 through D4 and the second set provides D5 through D8 from a second 8 x 4 array of memory devices.

The input data drivers shown in Figure 7 also represent four inverters, one for each data input bit, in the same manner as the address drivers and output selectors. The input data is distributed in common to both 8 x 4 memory arrays. The particular column of eight MCM4064's for which the data is intended is selected by the MC4007 data selectors which drive the write enable inputs. Only one of the eight MCM4064's in the columns will actually write in this data and is selected by the chip enable input. Thus only one device in the 8 x 4 array will actually accept the input data: that device chosen by the coincidence of the CE and W lines.

The entire system shown, therefore, constitutes a 512-word by 8-bit memory and requires the following devices:

64	MCM4064	64 bit memories
4	MC4000	One-of-four data selectors
1	MC4006	One-of-eight decoders
1	MC4007	One-of-four decoders
11	MC7404	Hex inverters

The system access time is the total of the input decoding times, memory access times, and output selection delay times. However, this may be decreased for read-read cycles by utilizing the maximum and minimum access times. Thus, if one changes the address or chip enable inputs every 60 ns, there is at least 15 ns of valid data on the memory output. Therefore, an output buffering scheme which can latch up the memory contents within 15 ns could be used to reduce the read-read access time to 60 ns plus the address/chip enable skew times which are invariably present. A similar technique could be used for read-write and write-write cycles.

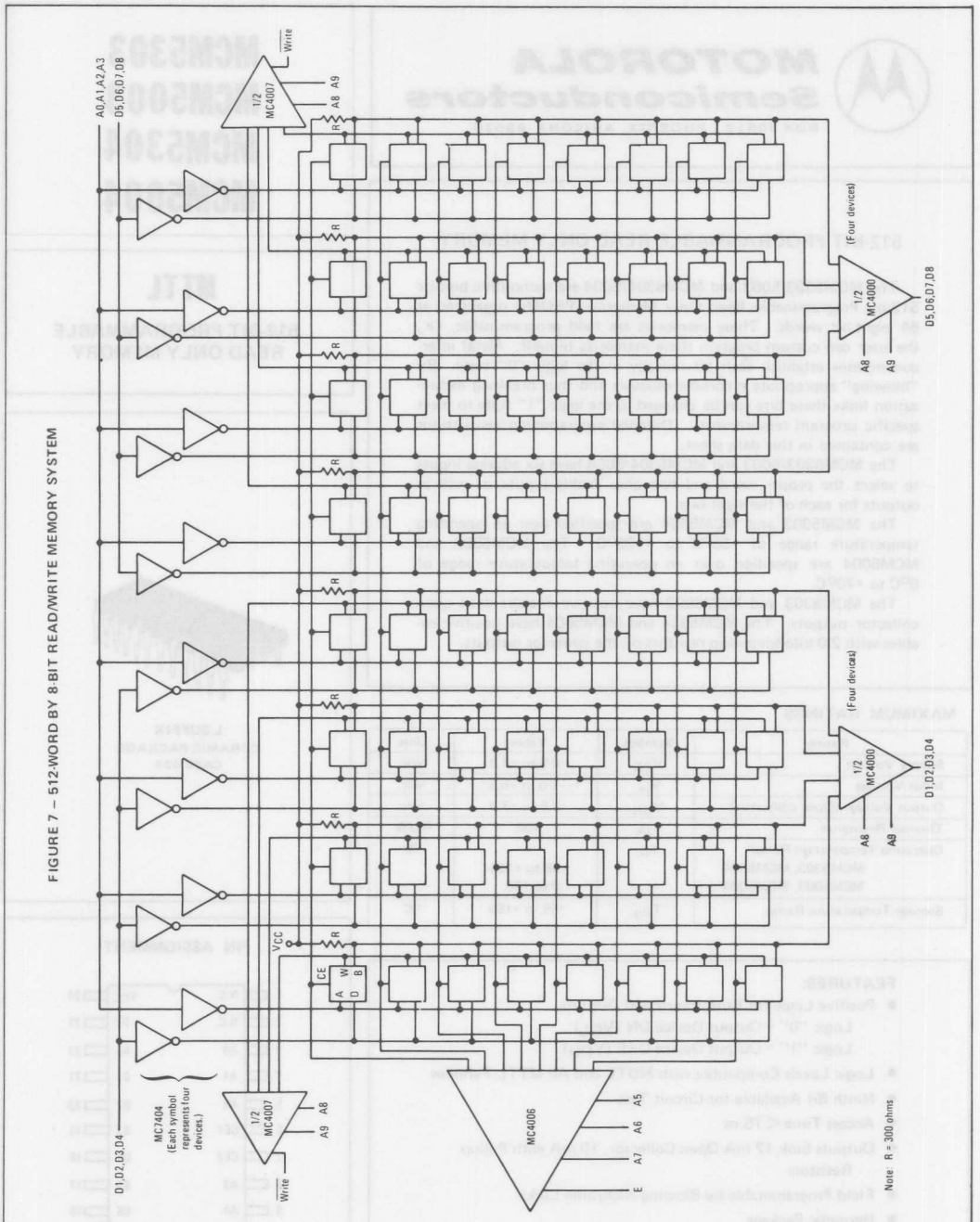


FIGURE 7 – 512-WORD BY 8-BIT READ/WRITE MEMORY SYSTEM

MOTOROLA Semiconductor Products Inc.
Printed in Switzerland

MOTOROLA
Semiconductors
BOX 20912 • PHOENIX, ARIZONA 85036

MCM5303
MCM5003
MCM5304
MCM5004

MTTL

512-BIT PROGRAMMABLE
READ ONLY MEMORY

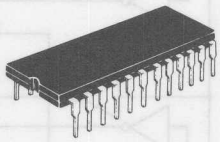
512-BIT PROGRAMMABLE READ ONLY MEMORY

The MCM5303/5003 and MCM5304/5004 are monolithic bipolar 512-bit Programmable Read Only Memories (PROMs) organized as 64 eight-bit words. These memories are field programmable, i.e., the user can custom program these memories himself. Metal interconnections establish each bit initially in the logic "0" state. By "blowing" appropriate nichrome resistors and thus breaking metalization links these bits can be changed to the logic "1" state to meet specific program requirements. Detailed programming instructions are contained in this data sheet.

The MCM5303/5003 and MCM5304/5004 have six address inputs to select the proper word and two chip enable inputs, as well as outputs for each of the eight bits.

The MCM5303 and MCM5304 are specified over an operating temperature range of -55°C to +125°C. The MCM5003 and MCM5004 are specified over an operating temperature range of 0°C to +70°C.

The MCM5303 and MCM5003 have positive enables with open collector outputs. The MCM5304 and MCM5004 have positive enables with 2.0 kilohm pullup resistors on the collector outputs.

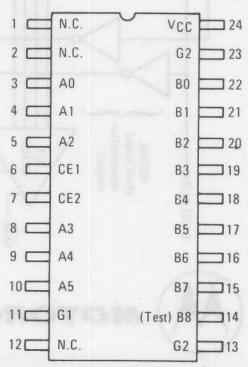


L SUFFIX
CERAMIC PACKAGE
CASE 623

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	-0.5 to +7.0	Vdc
Input Voltage	V_{in}	-1.0 to +5.5	Vdc
Output Voltage (Open collectors)	V_{OH}	-0.5 to +7.0	Vdc
Thermal Resistance	θ_{JA}	100	°C/W
Operating Temperature Range MCM5303, MCM5304 MCM5003, MCM5004	T_A	 -55 to +125 0 to +70	°C
Storage Temperature Range	T_{stg}	-55 to +165	°C

PIN ASSIGNMENT



FEATURES:

- Positive Logic for Both Inputs and Outputs
  Logic "0" = Output Device ON (V_{OL})
  Logic "1" = Output Device OFF (V_{OH})
- Logic Levels Compatible with MDTL and All MTTL Families
- Ninth Bit Available for Circuit Test
- Access Time < 75 ns
- Outputs Sink 12 mA Open Collector, 10 mA with Pullup Resistors
- Field Programmable by Blowing Nichrome Links
- Hermetic Package

APPLICATIONS:

- Look Up Tables
- Micro Programs
- Decode Functions
- Code Conversion
- Number Conversion
- Random Logic
- Character Generation

DC ELECTRICAL CHARACTERISTICS (T_A = -55°C to +125°C for MCM5303 and MCM5304, 0°C to +70°C for MCM5003 and MCM5004 unless otherwise noted)

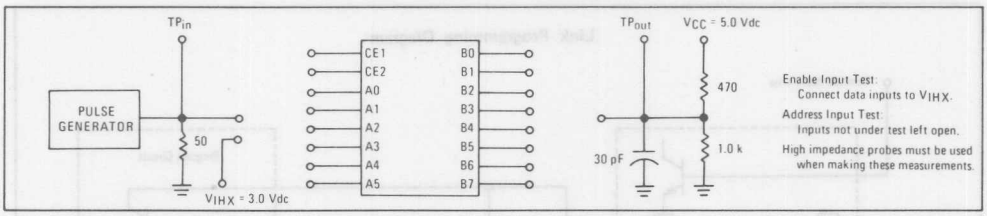
Characteristic		Symbol	Min	Max	Unit
Input Forward Current (V_{IL} = 0.4 Vdc, V_{CC} = 5.25 Vdc)		I_{IL}	–	1.6	mAdc
Input Leakage Current (V_{IH} = V_{CC} = 5.25 Vdc)		I_{IH}	–	100	μAdc
Logic "0" Output Voltage*		V_{OL}			Vdc
(T_A = 0°C to +125°C for MCM5303 and MCM5304, 0°C to +70°C for MCM5003 and MCM5004)					
(I_{OL} = 12 mAdc, V_{CC} = 4.75 Vdc)	Open Collectors		–	0.45	
(I_{OL} = 10 mAdc, V_{CC} = 4.75 Vdc)	Pullup Resistors		–	0.45	
(T_A = -55°C for MCM5303 and MCM5304)					
(I_{OL} = 12 mAdc, V_{CC} = 4.75 Vdc)	Open Collectors		–	0.50	
(I_{OL} = 10 mAdc, V_{CC} = 4.75 Vdc)	Pullup Resistors		–	0.50	
Logic "1" Output Voltage (I_{OH} = −0.5 mAdc, V_{CC} = 4.75 Vdc)	Pullup Resistors	V_{OH}	2.5	–	Vdc
Output Leakage Current (V_{CC} = V_{CEX} = 5.25 Vdc)	Open Collectors	I_{CEX}	–	200	μAdc
Power Supply Drain Current (Enable and all other inputs grounded, V_{CC} = 5.0 Vdc)	Open Collectors	I_{CC}	–	95	mAdc
	Pullup Resistors		–	120	

AC ELECTRICAL CHARACTERISTICS (V_{CC} 5.0 Vdc, T_A 25°C)

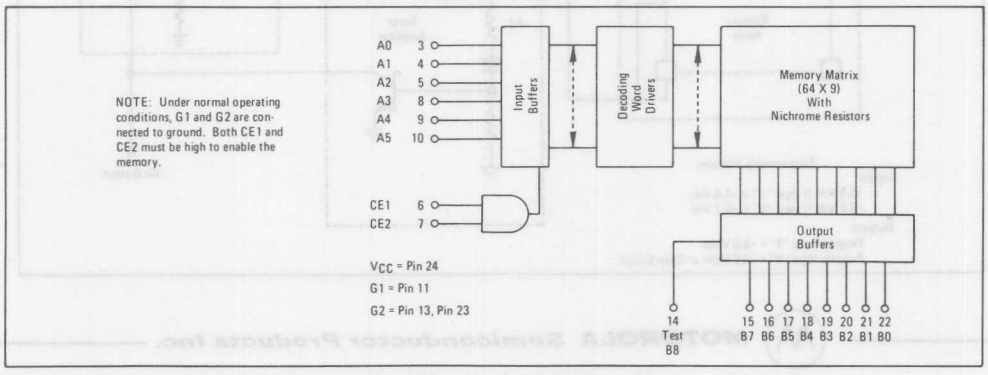
Access Times* (30pF Load)				ns
Address to Output	t_{AO}	25	75	
Enable to Output	t_{EO}	25	75	

*Pin 13 is schematically connected to G2. For optimum propagation delay and V_{OL} characteristics, externally tie Pin 13 to Pin 23 (G2).

SWITCHING TIME TEST CIRCUIT



BLOCK DIAGRAM



PROGRAMMING THE MCM5303/5003 AND MCM5304/5004

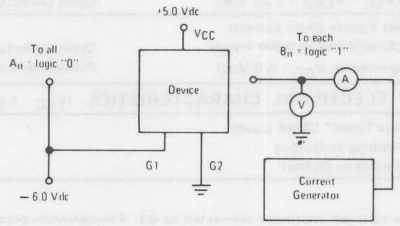
The table and diagram below give instructions for field programming the MCM5303/5003 and MCM5304/5004. All data given is for ambient temperatures of 25°C. If necessary, further programming aid can be obtained from Motorola engineering and product marketing personnel by contacting your nearest Motorola sales office.

Programming Voltage Limits

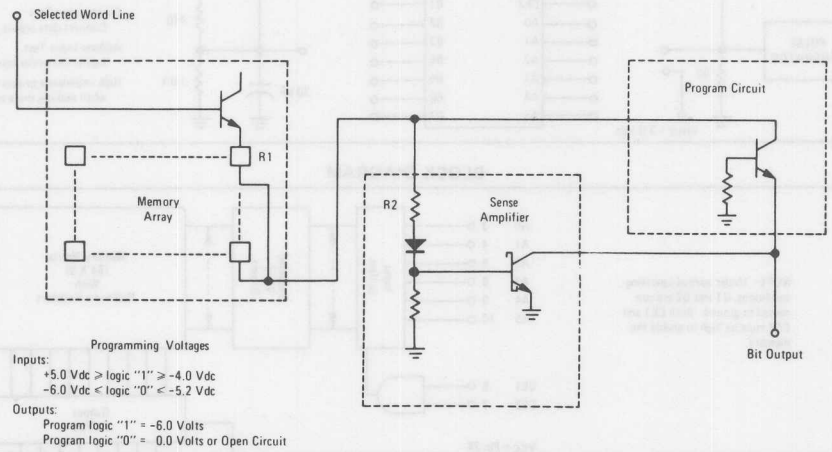
	Symbol	Value	Unit
Address and Chip Enable Voltages	V_{IH} V_{IL}	−4.0 to +5.0 −6.0 to −5.2	Vdc
Power Supply Voltage	V_{CC}	+5.0 ±5%	Vdc
G1 Voltage	V_{G1}	−6.0 ±5%	Vdc
G2 Voltage	V_{G2}	0.0	Vdc
Program Voltage at Desired Bit Output	V_{BP}	−6.0 ±5%	Vdc

Programming Procedure

1. Select the address code desired. Connect low (logic "0") inputs to −6.0 Vdc nominal. Leave high (logic "1") inputs unconnected.
2. With the output voltage of a 120-mA current generator clamped to −6.0 Vdc, apply a negative-going current pulse of 800 ms duration to any output to be programmed as a logic "1"
3. Repeat step 2 for each output to be programmed as a logic "1", one bit at a time.
4. Select next address code desired and repeat steps 2 and 3.



Link Programming Diagram



Programming Voltages

Inputs:
+5.0 Vdc ≥ logic "1" ≥ +4.0 Vdc
−6.0 Vdc ≤ logic "0" ≤ −5.2 Vdc

Outputs:
Program logic "1" = −6.0 Volts
Program logic "0" = 0.0 Volts or Open Circuit

MOTOROLA Semiconductor Products Inc.

TRUTH TABLE FORMAT

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
WORD 0								
WORD 1								
WORD 2								
WORD 3								
WORD 4								
WORD 5								
WORD 6								
WORD 7								
WORD 8								
WORD 9								
WORD 10								
WORD 11								
WORD 12								
WORD 13								
WORD 14								
WORD 15								
WORD 16								
WORD 17								
WORD 18								
WORD 19								
WORD 20								
WORD 21								
WORD 22								
WORD 23								
WORD 24								
WORD 25								
WORD 26								
WORD 27								
WORD 28								
WORD 29								
WORD 30								
WORD 31								

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
WORD 32								
WORD 33								
WORD 34								
WORD 35								
WORD 36								
WORD 37								
WORD 38								
WORD 39								
WORD 40								
WORD 41								
WORD 42								
WORD 43								
WORD 44								
WORD 45								
WORD 46								
WORD 47								
WORD 48								
WORD 49								
WORD 50								
WORD 51								
WORD 52								
WORD 53								
WORD 54								
WORD 55								
WORD 56								
WORD 57								
WORD 58								
WORD 59								
WORD 60								
WORD 61								
WORD 62								
WORD 63								

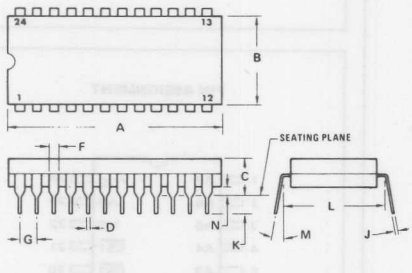
WHY THE NINTH BIT?

The ninth bit was designed into the MCM5303/MCM5003 and the MCM5304/MCM5004 because field-programmable ROMs present testing problems not encountered with conventional mask-programmable ROMs.

Three areas of testing are affected: Program Element Testing, Functional Testing, and AC Testing. The ninth bit helps to solve the problem of Program Element Testing by assuring that links can be blown without destroying any of the normal 64x8 bit array.

Functional and ac performance are assured by verifying that changes do occur at the outputs as the addresses change. This is important in that all of the outputs are in a logic "0" state regardless of the address selected, and no way is available to determine whether the functions are correctly operating without the ninth testing bit.

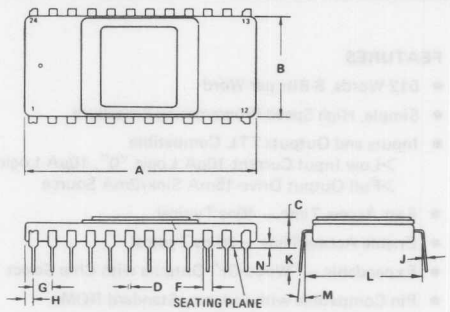
PACKAGE DIMENSIONS



DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES MAX
A	31.24	32.26	1.230	1.270
B	12.70	13.72	0.500	0.540
C	4.06	5.08	0.160	0.200
D	0.41	0.51	0.016	0.020
F	1.27	1.52	0.050	0.060
G	2.54 BSC		0.100 BSC	
J	0.20	0.30	0.008	0.012
K	2.92	3.43	0.115	0.135
L	15.37 BSC		0.605 BSC	
M	5°	15°	5°	15°
N	0.51	0.76	0.020	0.030

NOTES:
1. DIM "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.
2. LEADS WITHIN 0.13 mm (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION. (WHEN FORMED PARALLEL)

CASE 623



DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES MAX
A	29.34	30.86	1.155	1.215
B	12.70	14.22	0.500	0.560
C	3.05	3.94	0.120	0.155
D	0.38	0.51	0.015	0.020
F	0.89	1.40	0.035	0.055
G	2.54 BSC		0.100 BSC	
H	0.89	1.40	0.035	0.055
J	0.20	0.30	0.008	0.012
K	2.92	3.68	0.115	0.145
L	14.86	15.87	0.585	0.625
M	–	15°	–	15°
N	0.51	1.14	0.020	0.045

NOTES:
1. LEADS WITHIN 0.13 mm (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE WITH MAXIMUM MATERIAL CONDITION.
2. LEAD NO. 1 CUT FOR IDENTIFICATION, OR BUMP ON TOP.
3. DIM "L" TO INSIDE OF LEADS. (MEASURED 0.51 mm (0.020) BELOW PKG BASE)

CASE 684

MOTOROLA Semiconductor Products Inc.
Printed in Switzerland

MOTOROLA
Semiconductors
BOX 20912 • PHOENIX, ARIZONA 85036

MCM7640L,P
MCM7641L,P

Product Preview

BIPOLAR
512 X 8 BIT
PROGRAMMABLE
READ ONLY MEMORY

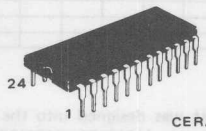
4096-BIT PROGRAMMABLE READ ONLY MEMORY

The MCM7640 (open collector output) and MCM7641 (three-state output) are monolithic bipolar 4096-bit Programmable Read Only Memories (PROMs) organized as 512 words by 8 bits per word. They are supplied with all bits storing a logical "1" (output high), and can be selectively programmed for a logical "0" (output low).

The MCM7640 and MCM7641 both use Motorola's standard fuse technology and are pin compatible with other industry standard PROMs.

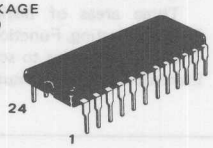
The field programmable PROM can be custom programmed to any pattern using a simple programming procedure. The Schottky Bipolar circuitry provides extremely fast access time.

In addition to the conventional storage array, one test row and two test columns are included to test programmability, and guarantee parametric and A.C. performance.



L SUFFIX
CERAMIC PACKAGE
CASE 623

P SUFFIX
PLASTIC PACKAGE
CASE 649



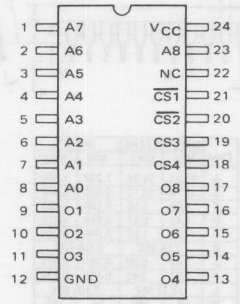
FEATURES

- 512 Words, 8-Bits per Word
- Simple, High Speed Programming Procedure
- Inputs and Outputs TTL Compatible
  - >Low Input Current-10μA Logic "0", 10μA Logic "1" (typical)
  - >Full Output Drive-15mA Sink/2mA Source
- Fast Access Time – 40ns Typical
- Enable Access Time – 20 ns Typical
- Expandable – "Wired-Or" Outputs with Chip Select
- Pin Compatible with Industry Standard ROMs
- Uses Harris Programming Cards

APPLICATIONS

- Bipolar Bit Slices
- Look-Up Tables
- Microprograms
- Decode Functions
- Code Conversion
- Number Conversion
- Random Logic
- Character Generation

PIN ASSIGNMENT



MOTOROLA
Semiconductors
BOX 20912 • PHOENIX, ARIZONA 85036

MCM7642L,P
MCM7643L,P

Product Preview

BIPOLAR
1024 X 4 BIT
PROGRAMMABLE
READ ONLY MEMORY

4096-BIT PROGRAMMABLE READ ONLY MEMORY

The MCM7642 (open collector output) and MCM7643 (three-state output) are monolithic bipolar 4096-bit Programmable Read Only Memories (PROMs) organized as 1024 words by 4 bits per word. They are supplied with all bits storing a logical "1" (output high), and can be selectively programmed for a logical "0" (output low).

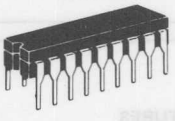
The MCM7660 and MCM7661 both use Motorola's standard fuse technology and are pin compatible with other industry standard PROMs.

The field programmable PROM can be custom programmed to any pattern using a simple programming procedure. The Schottky Bipolar circuitry provides extremely fast access time.

In addition to the conventional storage array, one test row and two test columns are included to test programmability, and guarantee parametric and A.C. performance.

P SUFFIX
PLASTIC PACKAGE
CASE 701-01

L SUFFIX
CERAMIC PACKAGE
CASE 726



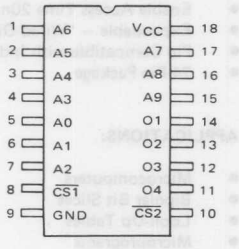
FEATURES

- 1024 Words, 4 Bits per Word
- Simple, High Speed Programming Procedure
- Inputs and Outputs TTL Compatible
  - Low Input Current-10μA Logic "0", 10μA Logic "1" (typical)
  - Full Output Drive – 15mA Sink/2mA Source
- Fast Access Time – 40ns Typical
- Enable Access Time 20ns Typical
- Expandable – "Wired-Or" Outputs with Chip Select
- Pin Compatible with Industry Standard ROMs
- Uses Harris Programming Cards
- 18 Pin Package

APPLICATIONS:

- Bipolar Bit Slices
- Look-Up Tables
- Microprograms
- Decode Functions
- Code Conversion
- Number Conversion
- Random Logic
- Character Generation

PIN ASSIGNMENT



MOTOROLA Semiconductors
BOX 20912 • PHOENIX, ARIZONA 85036

MCM7660L,P
MCM7661L,P

Product Preview

BIPOLAR
1024 X 8 BIT
PROGRAMMABLE
READ ONLY MEMORY

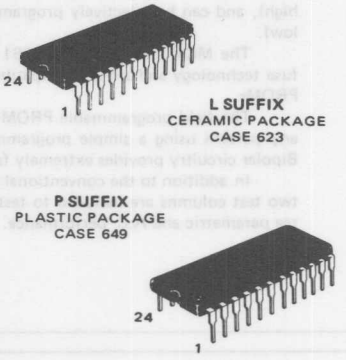
8192-BIT PROGRAMMABLE READ ONLY MEMORY

The MCM7660 (open collector output) and MCM7661 (three-state output) are monolithic bipolar 8192-bit Programmable Read Only Memories (PROMs) organized as 1024 words by 8 bits per word. They are supplied with all bits storing a logical "1" (output high), and can be selectively programmed for a logical "0" (output low).

The MCM7660 and MCM7661 both use Motorola's standard fuse technology and are pin compatible with other industry standard PROMs.

The field programmable PROM can be custom programmed to any pattern using a simple programming procedure. The Schottky Bipolar circuitry provides extremely fast access time.

In addition to the conventional storage array, one test row and two test columns are included to test programmability, and guarantee parametric and A.C. performance.



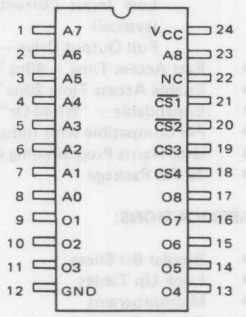
FEATURES

- Pin Compatible with Industry Standard EROMs and ROMs
- 1024 Words, 8-Bits per Word
- Simple, High Speed Programming Procedure
- Inputs and Outputs TTL Compatible
  - >Low Input Current-10μA Logic "0", 10μA Logic "1" (typical)
  - >Full Output Drive – 15mA Sink/2mA Source
- Fast Access Time – 40ns Typical
- Enable Access Time 20ns Typical
- Expandable – "Wired-Or" Outputs with Chip Select
- Pin Compatible with Industry Standard ROMs
- 24-Pin Package

APPLICATIONS:

- Microcomputers
- Bipolar Bit Slices
- Look-Up Tables
- Microprograms
- Decode Functions
- Code Conversion
- Number Conversion
- Character Generation

PIN ASSIGNMENT



EUROPEAN MOTOROLA SEMICONDUCTOR SALES OFFICES

DENMARK
Motorola A/S
Bredebovej 23
DK-2800 Lyngby
Tel. (02) 88 44 55

FRANCE
Motorola Semiconducteurs S.A.
Headquarter
15-17, avenue de Ségur
75007 Paris
Tel. 551 50 61
Sales Office
42, avenue de La Plaine-Fleurie
38240 Meylan (Grenoble)
Tel. (76) 90 61 32

WEST GERMANY
Motorola GmbH, Geschäftsbereich Halbleiter
Headquarter
Heinrich-Hertz-Strasse 1
D-6204 Taunusstein-Neuhof 5
Tel. (06128) 87 28 00
Sales Office Hannover
Hans-Böckler-Strasse 30
3012 Langenhagen
Tel. (0511) 77 20 37
Sales Office
Fiedlerstrasse 5
8 München 71
Tel. (089) 79 89 38
Sales Office
Virnsbergerstrasse 43
8500 Nürnberg
Tel. (0911) 6 57 61
Sales Office
Stralsunder Strasse 1
7032 Sindelfingen
Tel. (07031) 8 30 74/8 30 75

HOLLAND
Motorola B.V.
Emmalaan 41
Utrecht
Tel. (030) 51 02 07

ITALY
Motorola S.p.A.
Headquarter
Via Ciro Menotti 11
20129 Milano
Tel. 738 61 41/2/3
Sales Office
Via Portanova 10
40123 Bologna – Tel. 26 69 05
Sales Office
Via Costantino Maes 68
00162 Roma – Tel. 831 47 46

NORWAY
Motorola A/B.
Brugt. 1
Oslo 1 – Tel. (02) 41 91 40

SOUTH AFRICA
Motorola South Africa (Pty) Ltd.
Pretoria Road 639
Bramley 2018
Tel. (002711) 786 11 84

SWEDEN
Motorola AB.
Virebergsvaegen 19
17140 Solna – Tel. (08) 82 02 95

SWITZERLAND
Motorola Semiconductor Products S.A.
Alte Landstrasse 101
8702 Zollikon – Tel. (01) 65 56 56

UNITED KINGDOM
Motorola Ltd.
Headquarter
York House, Empire Way
Wembley Middlesex
Tel. (01) 902 88 36
Sales Office
10-12, Mount Street, Television House
Manchester M2 5WS, Lancs
Tel. (61) 833 07 31/833 07 34
Sales Office
Colvilles Road, Kelvin Estate
East Kilbride, Scotland
Tel. (3552) 39 101

HEADQUARTERS EUROPEAN OPERATIONS

SWITZERLAND
Motorola Inc.
Semiconductor Products Division
16, chemin de la Voie-Creuse, P.O. Box 8
1211 Genève 20
Tel. (022) 33 56 07

FRANCHISED MOTOROLA SEMICONDUCTOR DISTRIBUTORS

AUSTRIA
Elbatex GmbH
Breitenfurterstrasse 381
1235 Wien
Tel. (222) 86 91 58

BELGIUM
Diode Belgium
Rue Picard 202-204
1020 Bruxelles
Tel. (02) 428 51 05

DENMARK
GDS-Henckel ApS
Fyrrevangen 4
4622 Havdrup
Tel. (03) 38 57 16

EIRE
GDS (Sales) Limited
36, Pearce Street
Dublin 2
Tel. 78 22 32

FINLAND
Field Oy
Veneentekijantie 18
00210 Helsinki 21
Tel. (90) 69 22 577

FRANCE
Celdis S.A.
53, rue Charles-Frérot
94250 Gentilly
Tel. (01) 581 00 20
Ste F. Feutrier
Rue des Trois-Glorieuses
42270 St-Priest-en-Jarez (St-Etienne)
Tel. (1577) 74 67 33
Ets. Gros S.A.
14, avenue du Général-Leclerc
5400 Nancy
Tel. (1528) 24 24 22/32
Ets. Gros S.A.
13, rue Victor-Hugo
59350 St-André-lez-Lille
Tel. (1620) 55 64 70
S.C.A.I.B. S.A.
31, Quai Rambaud
69002 Lyon
Tel. 42 40 92
S.C.A.I.B. S.A.
15-17, avenue de Ségur
75007 Paris
Tel. (01) 555 95 54
S.C.A.I.B. S.A.
14, rue de Brest
35000 Rennes
Tel. (1699) 59 40 91
Sté. Commerciale Toutelectric
15-17, boulevard Bonrepos
31000 Toulouse
Tel. (1561) 62 47 84

GERMANY
Celdis GmbH
Orleansplatz 5
8000 München 80
Tel. (089) 45 43 06
Distron oHG
Mecklenburgische Strasse 24B
1000 Berlin 33
Tel. (030) 8 24 30 61
EBV Elektronik Vertriebs-GmbH
In der Meineworth 9A
3006 Burgwedel 1/Hannover
Tel. (051) 39 45 70
EBV Elektronik Vertriebs-GmbH
Oststrasse 129
4000 Düsseldorf
Tel. (0211) 8 48 46
EBV Elektronik Vertriebs-GmbH
Myliusstrasse 54
6000 Frankfurt
Tel. (0611) 72 04 16
EBV Elektronik Vertriebs-GmbH
Gabriel-Max-Strasse 72
8000 München 90
Tel. (089) 64 40 55
EBV Elektronik Vertriebs-GmbH
Alexanderstrasse 63
7000 Stuttgart 1
Tel. (0711) 24 74 81/82/83
Jermyn GmbH
Postfach 1146
6277 Camberg
Tel. (06434) 60 05
Jermyn GmbH
Rheinstrasse 7
8000 München 40
Tel. (089) 39 88 01
Mütron Müller & Co. KG
Bornstrasse 22
2800 Bremen
Tel. (0421) 31 04 85
RTG, E. Springorum KG
Bronnerstrasse 7
4600 Dortmund
Tel. (0231) 57 92 52
RTG, E. Springorum KG
Hamburgerstrasse 134
2000 Hamburg 76
Tel. (040) 29 29 66
RTG, E. Springorum KG
Vollstrasse 35
3000 Hannover-Vahrenwald
Tel. (0511) 66 20 26
RTG, E. Springorum KG
Halleschestrasse 1
6800 Mannheim-Vogelstang
Tel. (0621) 70 88 88
RTG, E. Springorum KG
Ungererstr. 43 – 8000 München 40
Tel. (089) 36 65 00
RTG, E. Springorum KG
Reutlingerstrasse 87
7000 Stuttgart-Degerloch
Tel. (0711) 76 64 28
RTG, E. Springorum KG
Mendelssohn-Bartholdy-Strasse 6
6200 Wiesbaden
Tel. (06121) 52 73 09
SASCO Vertrieb von elektronischen Bauelementen GmbH
Postfach 3066
4005 Düsseldorf/Meerbusch 3
Tel. (02150) 14 33
SASCO Vertrieb von elektronischen Bauelementen GmbH
Postfach 890214
3000 Hannover
Tel. (0511) 86 25 86
SASCO Vertrieb von elektronischen Bauelementen GmbH
Hermann-Oberth-Strasse 16
8001 Putzbrunn b. München
Tel. (089) 46 50 81
SASCO Vertrieb von elektronischen Bauelementen GmbH
Katharinenstrasse 18
7000 Stuttgart 1
Tel. (0711) 24 45 21
Technoprojekt
Ostring 150
6231 Schwalbach/Ts
Tel. (06196) 8 21 00
Technoprojekt
Heinrich-Ebner-Strasse 13
7000 Stuttgart – Bad Cannstatt
Tel. (0711) 56 17 12

GREECE
Macedonian Electronics Ltd.
Pontou 16 - Charibou
Thessaloniki
Tel. 42 60 32

HOLLAND
B.V. Diode
Hollantlaan 22 – Utrecht
Tel. (030) 88 42 14

IRAN
Milcom LTD, Motorola Building
Niloo Street, Vanak Square
Theran
Tel. 68 12 14/15

ISRAEL
Motorola Israel Ltd.
16, Kremenetzki Street
Tel Aviv
Tel. 36 941/42 – 38 973

ITALY
Celdis Italiana S.p.A. (Main Office)
Via Luigi Barzini 20
20125 Milano
Tel. (02) 68 89 651
Celdis Italiana S.p.A.
Via Turati 33
40055 Castenaso
Celdis Italiana S.p.A.
Via Lorenzo il Magnifico 109
00162 Roma
Tel. (06) 42 38 55/427 15 50
Celdis Italiana S.p.A.
Strada Maddalena 44/2
10020 Revigliasco
Cramer Italia S.p.A.
Via S. Simpliciano 2
20121 Milano
Tel. (02) 87 23 16/87 23 97
Cramer Italia S.p.A.
Via Malta 5
40135 Bologna
Tel. (051) 42 28 90
Cramer Italia S.p.A.
Via Cristoforo Colombo 134
00147 Roma
Tel. (06) 513 30 41/513 93 90
Cramer Italia S.p.A.
Corso Traiano 28/15
10135 Torino
Tel. (011) 619 20 62/619 20 67

NORWAY
Ole Tandberg Elektro A/S
Skedsmogaten 25 – Oslo 6
Tel. (02) 19 70 30

POLAND
PHZ Transpol S.A. (Intraco Building)
Ul. Stawki 2
00-950 Warsaw 1
Tel. (004822) 33 29 11

PORTUGAL
Equipamentos de Laboratorio LDA
Rua Pedro Nunes 47
Lisbon 1
Tel. 97 02 51

SOUTH AFRICA
L'Electron
704 Pretoria Main Road
Wynberg, TVL
Tel. 40-62 75 – 40-62 96/7/8/9

SPAIN
Hispano Electronics S.A. (Main Office)
Commandante Zorita 8
Madrid 20
Tel. (01) 233 31 00/233 47 00
Hispano Electronics S.A.
Figols, 27-29
Barcelona 14 – Tel. 259 05 22/23
Hispano Electronics S.A.
Zabalbide, 42
Bilbao 6 – Tel. 423 83 09/423 80 62
Hispano Electronics S.A.
Navarro Reverter, 2
Valencia – Tel. 373 14 97

SWEDEN
Interelko AB.
Sandsborgsvägen 55
12233 Enskede – Tel. (08) 49 25 05

SWITZERLAND
Elbatex AG
Alb. Zwyssig-Strasse 28
5430 Wettingen
Tel. (056) 26 56 41
GDS (Sales) S.A.
3, rue de l'Aubépine
1211 Genève 9
Tel. (022) 21 59 77
GDS (Sales) S.A.
Mühlebachstrasse 54
8008 Zürich
Tel. (01) 47 28 50
Omni Ray AG
Dufourstrasse 56
8008 Zürich
Tel. (01) 34 07 66/32 93 70

TURKEY
Altay Kollektif Sirketi
Kizilirmak Sokak 2 Bakanlikar
Ankara
Tel. 25 25 06/07
ERA, Elektronik Sanayii Ve Ticaret A.S.
Eski Büyükdere Caddesi 49A, 4 Levent
Istanbul
Tel. 64 65 00/64 65 01

UNITED KINGDOM
Celdis Ltd.
37-39 Loverock Road
Reading, Berks, RG3, 1ED
Tel. (0734) 58 22 11
Cramer Components Ltd
16, Uxbridge Road
Ealing, London W.5 2BP
Tel. (01) 579 30 01
GDS (Sales) Limited
Michaelmas House, Salt Hill, Bath Road
Slough,
Tel. (75) 30 211
ITT Electronic Services
Edinburgh Way
Harlow, Essex CM20 2DF
Tel. Harlow (0279) 26 777
Jermyn Distribution
Vestry Estate
Sevenoaks, Kent
Tel. (0732) 5 01 44
Lock Distribution
Division of A.M. Lock & Co. Ltd.
Neville Street
Oldham OL9 6LF, Lancs
Tel. (061) 652 04 31
Semicomps Ltd.
Wellington Road/London Colney
St Albans, Herts AL2 1EZ
Tel. Bowmans Green (61) 245 22

YUGOSLAVIA
Elektrotehna
Karadjardjeva 71
11000 Beograd
Tel. 62 81 55/63 13 55

EUROPEAN SEMICONDUCTOR FACTORIES

UNITED KINGDOM
Motorola Semiconductors Ltd.
Colvilles Road, Kelvin Estate
East Kilbride/Glasgow (Scotland)
Tel. (3552) 39 101

FRANCE
Motorola Semiconducteurs S.A.
Canto Laouzetto – Le Mirail
31023 Toulouse CEDEX
Tel. (1561) 40 11 88

EUROPEAN MOTOROLA SEMICONDUCTOR SALES OFFICES

FRANCHISED MOTOROLA SEMICONDUCTOR DISTRIBUTORS

EUROPEAN SEMICONDUCTOR FACTORIES